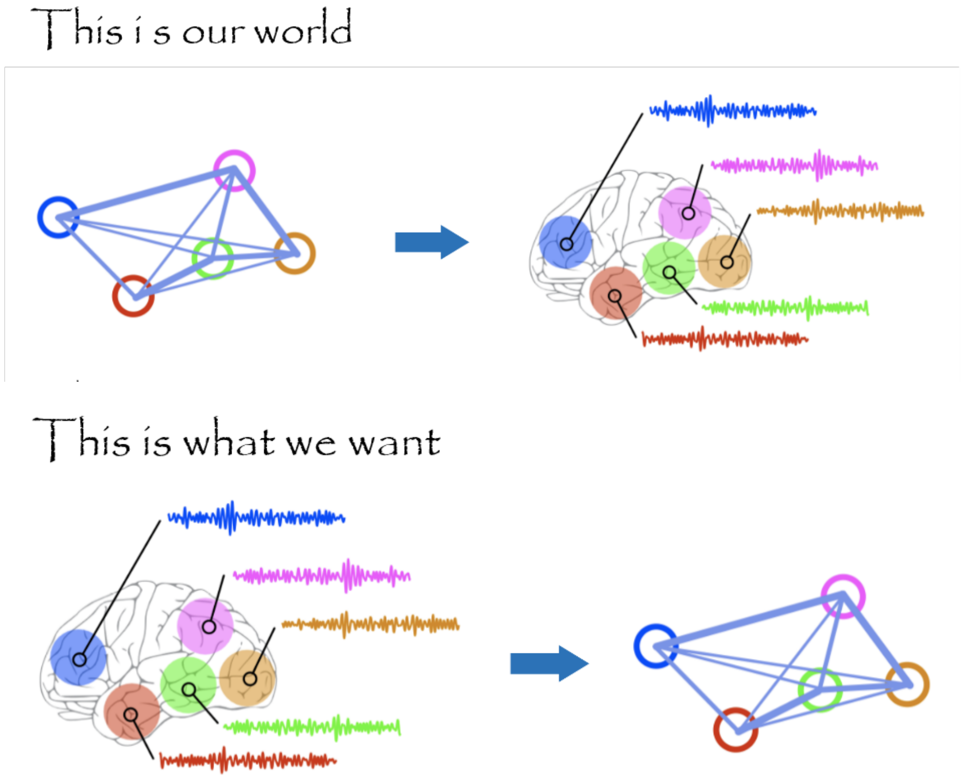
Circuits and Systems
Mekelweg 4,
2628 CD Delft
The Netherlands
https://sps.ewi.tudelft.nl/

SPS-2024-00

M.Sc. Thesis

Graph Topology Identification Based on Covariance Matching

Yongsheng Han B.Sc.



[1]

[1] modified based on [1]

Faculty of Electrical Engineering, Mathematics and Computer Science
Delft University of Technology

Graph Topology Identification Based on Covariance Matching

A New View on Network Analysis

Thesis

submitted in partial fulfillment of the
requirements for the degree of

Master of Science

in

Electrical Engineering

by

Yongsheng Han B.Sc.
born in Rizhao, China

This work was performed in:

Circuits and Systems Group
Department of Microelectronics
Faculty of Electrical Engineering, Mathematics and Computer Science
Delft University of Technology

Delft University of Technology

Delft University of Technology
Department of
Microelectronics

The undersigned hereby certify that they have read and recommend to the Faculty of Electrical Engineering, Mathematics and Computer Science for acceptance a thesis entitled **"Graph Topology Identification Based on Covariance Matching"** by **Yongsheng Han B.Sc.** in partial fulfillment of the requirements for the degree of **Master of Science**.

Dated: My Graduation Date

Chairman: ______________________________

prof.dr.ir. G.J.T. Leus

Advisor: ______________________________

dr.ir. A. Natali

Committee Members: ______________________________

prof.dr.ir. E. Isufi

____

Abstract

Graph signal processing (GSP) extends classical signal processing to signals on graphs, enabling the analysis of complex data structures through graph theory. A core challenge in GSP is graph topology identification, which aims to deduce the graph structure that best explains observed signal dependencies.

This project addresses graph topology identification for applications where the underlying structure of systems like brain and social networks is not directly observable. Traditional approaches based on signal matching and spectral templates have limitations, particularly in handling scale issues and sparsity assumptions. We introduce a novel covariance matching methodology that efficiently reconstructs the graph topology using observable data. For the structural equation model (SEM) using an undirected graph, we demonstrate that our method can converge to the correct result under relatively soft conditions. Furthermore, we extend our methodology to polynomial models, sparse directed graphs, and any known Gaussian distribution of latent variables, broadening its applicability and utility in diverse graph-based systems. Experimental results demonstrate that our method outperforms existing techniques, offering new directions for future research in graph topology identification.

Acknowledgments

Reflecting on my journey, the term ”research,” a reiteration of the quest for knowledge, perfectly encapsulates my experience with this thesis. It has been an incredibly enjoyable and immersive journey, and I am profoundly grateful for the opportunity and guidance provided by Geert. His high-level oversight and timely suggestions not only shaped my path but also anticipated challenges and milestones, many of which came to pass. Thank you, Geert, for your experienced and insightful mentorship.

I am equally thankful to Alberto for his valuable advice and guidance throughout this process. His input has been instrumental in refining my work.

A special thank you to Elvin, whose participation in my thesis committee was immensely appreciated. His course on graph machine learning sparked my initial interest in this field, and his continued engagement has been inspiring.

I extend my heartfelt thanks to those who discussed math with me in community. I am especially grateful to Nikita Lisitsa for proposing the idea of using Cholesky decomposition and to Dan Fulea for his rigorous proof of Lemma 4.

To my family, your understanding and support have been the bedrock of my personal and academic life. I love you all dearly and am endlessly appreciative of everything you do.

As I write down these words on the occasion of the Mid-Autumn Festival, I am reminded of the enduring bonds we share, despite the distances that may separate us. May we all enjoy longevity and continue to share in each other’s lives, no matter the miles apart. (但愿人长久，千里共婵娟)

Written on the Mid-Autumn Festival

Yongsheng Han B.Sc.
Delft, The Netherlands

Nomenclature

The next list describes several symbols that will be later used within the body of the thesis

Graph

$\mathbf{S}$	Graph shift operator
$\mathbf{y}$	One sample
$\mathbf{Y}$	Multiple samples
T	Sample length
$\mathbf{C}_{\mathbf{y}}$	Sample covariance of $\mathbf{y}$, calculated as $\frac{\mathbf{Y}\mathbf{Y}^\top}{T}$
$\mathbf{\Sigma}_{\mathbf{y}}$	Ensemble covariance of $\mathbf{y}$, $\mathbb{E}\{\mathbf{y}\mathbf{y}^\top\}$
$\mathbf{m}_{\mathbf{y}}$	Sample mean of $\mathbf{y}$, calculated as $\frac{\mathbf{Y}\mathbf{1}}{T}$
$\boldsymbol{\mu}_{\mathbf{y}}$	Ensemble mean of $\mathbf{y}$, $\mathbb{E}\{\mathbf{y}\}$
$h(\cdot)$	A polynomial function
$\mathbf{H}$	For SEM: $\mathbf{H} = (\mathbf{I} - \mathbf{S})^{-1}$, for polynomial model: $\mathbf{H} = h(\mathbf{S})$

Linear algebra

$(\cdot)^n$	Matrix power for matrices, element-wise power for vectors
$(\cdot)^{-1}$	Inverse of matrix $(\cdot)$
$(\cdot)^{-\top}$	First inverse then transpose of matrix $(\cdot)$
$(\cdot)^{-n}$	First inverse then power of $(\cdot)$
$\mathbf{U}\,\mathrm{diag}(\boldsymbol{\lambda})\mathbf{U}^\top$	Eigenvalue decomposition (EVD) of symmetric matrices
$\mathbf{U}\,\mathrm{diag}(\boldsymbol{\lambda})\mathbf{U}^{-1}$	Eigenvalue decomposition (EVD) for any square matrix if it exists
$\mathbf{U}\,\mathrm{diag}(\boldsymbol{\lambda})\mathbf{V}^\top$	Singular value decomposition (SVD)

Optimization problem

$(\cdot)$	True value of variables
$\hat{(\cdot)}$	Estimated variable
$(\cdot)^*$	Optimal value of optimization variable

Contents

List of Figures

Introduction 1

1.1 Motivation and Contributions

Graph signal processing (GSP) extends classical signal processing concepts to signals defined on graphs, allowing the exploration of complex data structures through the lens of graph theory. A fundamental challenge in GSP is the identification of the graph topology from observed signals. This involves determining the underlying graph structure that most effectively explains the signal correlations or dependencies. This process is illustrated in Fig. 1.1.

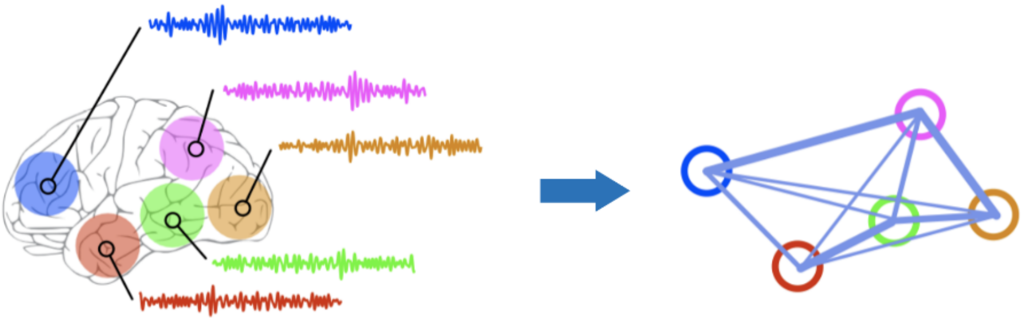


Figure 1.1: Illustration of uncovering hidden graph structures from observed data.[1]

Graph topology identification remains a critical issue in GSP, where systems are modeled as networks, yet their actual underlying structure is often invisible. Examples of such systems include brain functional connectivity networks and social networks. In these applications, while the direct graph structure is not observable, nodal data is typically available. For instance, in academic networks [3], the advisor-advisee links may not be visible, yet we can analyze collaborative patterns to uncover these connections. Similarly, in brain networks [4], neural signals provide indirect clues about the connectivity. Therefore, the primary challenge in graph topology identification lies in deducing the hidden graph structure from these nodal observations, a task that is fundamental for analyzing and understanding the interactions within these networks.

The structural equation model (SEM) is a popular tool to link nodal data with the graph, and it has been frequently used in graph topology identification [5, 6, 7, 8]. A remarkable result was provided by [8], where it was demonstrated that for sparse directed acyclic graphs (DAGs), the graph can be uniquely determined when the unknown latent variable is Gaussian with equal variance.

This foundational work spurred further developments, leading to more efficient algorithms as evidenced by [9] and [10]. However, for undirected graphs, methods using signal matching have performed poorly without the presence of exogenous variables, even with the introduction of sparsity constraints [11, 12]. On the other hand, spectral template-based approaches, such as the polynomial graphical lasso (PGL) algorithm, have shown potential in handling certain graphs [2], but the results typically differ from the true structure by a scale factor and require extensive restrictions (sparsity and sign) on the graph for the method to be effective.

To the best of our knowledge, no existing work has exhaustively addressed topology identification for undirected graphs using a SEM. This paper fills this gap by proposing a novel covariance matching-based method. We will prove that, under relatively soft conditions, our proposed method consistently converges to the correct result without encountering the scale issue often associated with other approaches. Additionally, our method is robust and does not require any assumptions about sparsity.

Even if we overlook the various issues with existing methods, once we consider non-white Gaussian latent variables, all current approaches fail. However, non-white latent variables often more closely reflect real-world situations. Current research, such as that in [13], suggests using multiple systems to estimate $\mathbf{S}$, which is highly impractical. In this paper, we also address this issue. In fact, when considering non-white Gaussian latent variables, our covariance matching approach maintains the same structure.

Regarding directed graphs, as we have mentioned before, all research to date has focused only on DAGs without considering cycles. In this paper, we aim to tackle this exciting and challenging problem. Our experiments have been very successful, and similar to the undirected case, we also extend our approach to non-white Gaussian latent variables.

Furthermore, if we consider more complex models, such as the polynomial model, existing methods either assume no known polynomial information, like in [2, 14], or can only handle very trivial polynomial information, as seen in [15]. However, in reality, we often encounter situations where we have some knowledge of the polynomial and it is not a trivial polynomial model. This brings us to the final part of our thesis, where we extend our method to handle polynomial models and non-white Gaussian latent variables.

Table 1.1: Overview of Our Method's Applicability

Model Type	**Directed**	**Sparse**	**Latent Variable Distribution**	**Applicability**
SEM	No	–	$\mathcal{N}(\mathbf{0}, \mathbf{\Sigma})$	Applicable
SEM	Yes	No	$\mathcal{N}(\mathbf{0}, \mathbf{\Sigma})$	Unsolvable
SEM	Yes	Yes	$\mathcal{N}(\mathbf{0}, \mathbf{\Sigma})$	Applicable
PM	No	–	$\mathcal{N}(\boldsymbol{\mu}, \mathbf{\Sigma}), \boldsymbol{\mu} \neq \mathbf{0}$	Applicable
PM	No	–	$\mathcal{N}(0, \mathbf{I})$	Applicable
PM	Yes	–	–	Unknown

In Table 1.1, we summarize the applicability of our covariance matching technique. The

"Sparse" column indicates whether the graph is sparse, whereas "Directed" denotes whether the graph is directed; '–' signifies that the condition does not need to be met for our method. For the SEM, we have nearly addressed most scenarios. For the polynomial model (PM), we have managed to handle the undirected configurations.

Our final experimental results demonstrate that our approach outperforms many existing methods and shows substantial potential for addressing more complex issues in future extensions. Overall, the covariance matching approach introduces new possibilities and directions within the field of graph topology identification.

1.2 Thesis Structure

The foundational approach of this study begins with the SEM, a framework that models influences among nodes using linear relations. In Chapter 2, we review existing literature to unify and reinterpret previous findings, setting the stage for our subsequent investigations.

Starting from the signal matching method, in Chapter 3, we try to apply this methodology to undirected graphs. We adopt the assumption of a zero mean white Gaussian distribution for the latent variables from prior work but demonstrate its limitations in achieving accurate results through theoretical proofs. Subsequently, we revisit the problem from a novel point of view and introduce our covariance matching approach. We further provide evidence that our method is effective for a substantial portion of graphs. At last, we drop the whiteness assumption and extend our method to any zero mean Gaussian distribution.

In Chapter 4, we address the challenges associated with directed graphs. The chapter begins by outlining the complexities of the problem and then explores strategies to exploit sparsity. We formulate the problem based on covariance matching and achieve an optimization problem over unitary matrices. To tackle this, we present a convex relaxation method for unitary matrices.

Chapter 5 starts from the unitary optimization framework discussed in Chapter 4 and introduces a more generalized approach. It builds on the methodologies developed in Chapter 4 by implementing an iterative method that encourages the solution variables to closely resemble unitary matrices.

In Chapter 6, we explore polynomial models, specifically focusing on undirected graphs. We start with the assumption that the latent variables follow a zero mean white Gaussian distribution. We extend the semidefinite relaxation method traditionally used for binary quadratic problems (BQPs) to solve optimization problems associated with polynomial models, i.e., discrete quadratic problems (DQPs). Eventually, we generalize our approach to accommodate any non-zero mean Gaussian distributed latent variables.

In Chapter 7, we summarize our findings and discuss potential directions for future research, providing a roadmap for extending our methodologies to new and emerging challenges.

Background 2

2.1 Graph Signal Processing

Graph signal processing (GSP) is the study of how to analyze and process data associated with graphs. Graphs are mathematical structures used to model pairwise relations between objects. In the context of GSP, a graph consists of nodes (or vertices) and edges (or links) connecting them. Nodes can be considered as abstract representations of real-world entities, while edges represent the relationships between these entities. In GSP, the term signal refers to any set of data values that are assigned to the nodes of the graph. These signals might represent measurements, observations, or attributes associated with each node, depending on the specific application and context.

2.1.1 Mathematical Representations of Graphs and Graph Signals

We often describe a graph with the following notation:

$$\mathcal{G} = \{\mathcal{V}, \mathcal{E}, \mathbf{S}\},$$

where $\mathcal{V} = \{1, \ldots, N\}$ represents the set of vertices, $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$ denotes the set of edges, and $\mathbf{S}$ is referred to as the graph shift operator (GSO). It can be considered as an $N \times N$ matrix that captures the graph topology and will be explained next. Each node $i \in \mathcal{V}$ is associated with a scalar value x_i. By stacking these values into a vector

$$\mathbf{x} = [x_1, \ldots, x_N]^\top \in \mathbb{R}^N,$$

we obtain what is known as a graph signal.

Similar to traditional signal processing, in GSP, we define the process generating graph signals as independent and identically distributed (i.i.d.). This assumption means that each signal observation is statistically independent from others and all share the same probability distribution. After capturing observations at T different instances $\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_T$, we stack these signals into a matrix given by

$$\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_T],$$

to represent the i.i.d. graph signals across these T samples.

2.2 Graph Shift Operator

In this section, we discuss some common forms of the matrix $\mathbf{S}$ that are essential for graph topology analysis. These representations facilitate various computational and analytical tasks related to graphs.

2.2.1 Adjacency Matrix

The adjacency matrix is a fundamental representation in graph theory, typically denoted by $\mathbf{A}$ for binary graphs and $\mathbf{W}$ for weighted graphs. In the binary adjacency matrix $\mathbf{A}$, the element A_{ij} is either 0 or 1; it is 1 if there is an edge from node j to node i, implying that $(j, i) \in \mathcal{E}$, and 0 otherwise. This indicates the mere existence of a connection between nodes. Conversely, in the weighted adjacency matrix $\mathbf{W}$, the element W_{ij} can take any real value, which represents the strength or capacity of the connection from node j to i.

Additionally, whether an adjacency matrix is symmetric or non-symmetric informs about the nature of the graph it represents. A symmetric adjacency matrix, where $A_{ij} = A_{ji}$ (or $W_{ij} = W_{ji}$), indicates an undirected graph where edges have no orientation. In contrast, a non-symmetric adjacency matrix represents a directed graph, where the presence of an edge from node i to node j does not necessarily imply the presence of an edge from node j to node i.

Remark 1. In GSP, it is important to note that the element A_{ij} actually represents a connection from node j to node i. This is opposite to the typical representation in graph theory, where A_{ij} denotes a link from node i to node j.

The main reason for this difference arises from how we model signal diffusion on graphs. If a signal $\mathbf{x}$ diffuses 1 hop on the graph to produce $\mathbf{y}$, and we wish to describe this relationship using $\mathbf{y} = \mathbf{A}\mathbf{x}$, then, due to matrix-vector multiplication, the i-th column of $\mathbf{A}$ must be associated with the elements of the i-th row of $\mathbf{x}$. This results in the i-th column of $\mathbf{A}$ actually representing outgoing connections from node i. To maintain the commonly used relation $\mathbf{y} = \mathbf{A}\mathbf{x}$ in signal processing, such adjustments are made in the definition of $\mathbf{A}$ in GSP.

2.2.2 Laplacian Matrix

Next, we consider the Laplacian matrix, typically denoted by $\mathbf{L}$. The Laplacian matrix is generally defined for undirected graphs and it is written as:

$$\mathbf{L} = \mathbf{D} - \mathbf{W}, \tag{2.1}$$

where $\mathbf{D}$ is the degree matrix, a diagonal matrix where each diagonal element D_{ii} is the sum of the weights of all edges connected to vertex i. This formulation reflects that $\mathbf{D} = \mathrm{Diag}(\mathbf{W}\mathbf{1})$, simplifying the understanding that $\mathbf{D}$ essentially captures the degree of each node. The Laplacian matrix plays a pivotal role in modeling diffusion processes on graphs and is intimately related to the smoothness of graph signals.

An important property of the Laplacian matrix is that all its eigenvalues are non-negative, and there is one eigenvalue equal to zero. This can be demonstrated through a simple proof.

Consider the quadratic form of the Laplacian:

$$\mathbf{x}^\top \mathbf{L}\mathbf{x} = \mathbf{x}^\top (\mathbf{D} - \mathbf{W})\mathbf{x} = \mathbf{x}^\top \mathbf{D}\mathbf{x} - \mathbf{x}^\top \mathbf{W}\mathbf{x}. \tag{2.2}$$

The term $\mathbf{x}^\top\mathbf{D}\mathbf{x}$ can be expanded to:

$$\mathbf{x}^\top\mathbf{D}\mathbf{x} = \sum_{i=1}^{N} D_{ii}x_i^2, \tag{2.3}$$

where D_{ii} is the degree of node i, and x_i is the i-th component of $\mathbf{x}$. The term $\mathbf{x}^\top\mathbf{W}\mathbf{x}$ sums the products of the adjacent entries, reflecting the connection strengths:

$$\mathbf{x}^\top\mathbf{W}\mathbf{x} = \sum_{i\neq j} W_{ij}x_ix_j. \tag{2.4}$$

Rearranging terms gives:

$$\mathbf{x}^\top\mathbf{L}\mathbf{x} = \frac{1}{2}\sum_{i\neq j} W_{ij}(x_i - x_j)^2. \tag{2.5}$$

This sum is clearly non-negative since each term $(x_i - x_j)^2$ is non-negative. Thus, all eigenvalues of $\mathbf{L}$ are non-negative. Furthermore, $\mathbf{L}\mathbf{1} = \mathbf{0}$ demonstrates that the vector of all ones, $\mathbf{1}$, is an eigenvector corresponding to the zero eigenvalue, proving that there is one eigenvalue equal to zero.

In fact, the expression $\mathbf{x}^\top\mathbf{L}\mathbf{x}$ also quantifies the total variation of the signal $\mathbf{x}$ across the graph. Specifically, it measures the degree to which $\mathbf{x}$ varies along the edges of the graph, making it a measure of the signal's smoothness. In GSP, a smaller value of $\mathbf{x}^\top\mathbf{L}\mathbf{x}$ implies that the signal varies minimally between connected nodes, thus indicating a smoother signal across the graph. Several papers consider this property of the Laplacian and optimize $\mathbf{x}^\top\mathbf{L}\mathbf{x}$ to find a graph over which the data is smooth [16, 17].

2.2.3 Normalized Laplacian Matrix

Finally, the normalized Laplacian matrix, denoted by $\tilde{\mathbf{L}}$, is defined as:

$$\tilde{\mathbf{L}} = \mathbf{D}^{-1/2}\mathbf{L}\mathbf{D}^{-1/2}, \tag{2.6}$$

where $\mathbf{D}$ is the diagonal degree matrix as previously defined. This normalization scales the Laplacian matrix by the degrees of the nodes, facilitating an invariant analysis relative to the graph's scale or the distribution of node degrees. The normalized Laplacian is especially beneficial in machine learning applications, such as spectral clustering and semi-supervised learning on graphs, where the characteristics of eigenvalues and eigenvectors of $\tilde{\mathbf{L}}$ play a crucial role.

The eigenvalues of the normalized Laplacian matrix, $\tilde{\mathbf{L}}$, are bounded between 0 and 2. We prove this next. Let $\mathbf{x}$ represent a normalized vector such that $\mathbf{x}^T\mathbf{x} = 1$, where each component x_i denotes the i-th element of $\mathbf{x}$. Let d_i represent D_{ii}. And let us rewrite the normalized Laplacian matrix $\tilde{\mathbf{L}}$ as

$$\begin{aligned}\tilde{\mathbf{L}} &= \mathbf{D}^{-1/2}\mathbf{L}\mathbf{D}^{-1/2} = \mathbf{D}^{-1/2}(\mathbf{D} - \mathbf{A})\mathbf{D}^{-1/2} \\ &= \mathbf{I} - \mathbf{D}^{-1/2}\mathbf{A}\mathbf{D}^{-1/2}.\end{aligned} \tag{2.7}$$

Then we can express $\mathbf{x}^T\tilde{\mathbf{L}}\mathbf{x}$ as

$$\begin{aligned}\mathbf{x}^T\tilde{\mathbf{L}}\mathbf{x} &= \mathbf{x}^T(\mathbf{I}-\mathbf{D}^{-1/2}\mathbf{A}\mathbf{D}^{-1/2})\mathbf{x} \\ &= \sum_{i\in\mathcal{V}} x_i^2 - \sum_{(i,j)\in\mathcal{E}} \frac{2x_ix_j}{\sqrt{d_id_j}} \\ &= \sum_{(i,j)\in\mathcal{E}} \left(\frac{x_i}{\sqrt{d_i}} - \frac{x_j}{\sqrt{d_j}}\right)^2 \\ &\geq 0.\end{aligned} \tag{2.8}$$

To show that $\mathbf{x}^T\tilde{\mathbf{L}}\mathbf{x} \leq 2$ we can derive:

$$\begin{aligned}\mathbf{x}^T\tilde{\mathbf{L}}\mathbf{x} &= \mathbf{x}^T(\mathbf{I}-\mathbf{D}^{-1/2}\mathbf{A}\mathbf{D}^{-1/2})\mathbf{x} \\ &= 2\sum_{i\in\mathcal{V}} x_i^2 - \sum_{i\in\mathcal{V}} x_i^2 - \sum_{(i,j)\in\mathcal{E}} \frac{2x_ix_j}{\sqrt{d_id_j}} \\ &= 2\sum_{i\in\mathcal{V}} x_i^2 - \sum_{(i,j)\in\mathcal{E}} \left(\frac{x_i}{\sqrt{d_i}} + \frac{x_j}{\sqrt{d_j}}\right)^2 \\ &\leq 2.\end{aligned} \tag{2.9}$$

Since the largest eigenvalue of $\tilde{\mathbf{L}}$, denoted as $\lambda_{\max}(\tilde{\mathbf{L}})$, and the smallest eigenvalue of $\tilde{\mathbf{L}}$, denoted as $\lambda_{\min}(\tilde{\mathbf{L}})$, satisfy the following properties [18]:

$$\max_{\mathbf{x}^T\mathbf{x}=1} \mathbf{x}^T\tilde{\mathbf{L}}\mathbf{x} = \lambda_{\max}(\tilde{\mathbf{L}}) \tag{2.10}$$

$$\min_{\mathbf{x}^T\mathbf{x}=1} \mathbf{x}^T\tilde{\mathbf{L}}\mathbf{x} = \lambda_{\min}(\tilde{\mathbf{L}}) \tag{2.11}$$

we can conclude our proof.

2.3 Graph Topology Identification

Graph topology identification (GTI), also known as graph learning or learning graphs from data, refers to the task of inferring the relationships between nodes based on observations of nodal data. Specifically, given T observations across N nodes organized in a data matrix $\mathbf{X} \in \mathbb{R}^{N\times T}$, and equipped with prior knowledge such as distribution assumptions or data models, the objective is to construct a graph $\mathcal{G} = \{\mathcal{V}, \mathcal{E}, \mathbf{S}\}$. This graph depicts the relationships among the variables. Each column of the matrix $\mathbf{X}$ is interpreted as a signal on the graph, based on the node configuration of graph $\mathcal{G}$.

2.3.1 Challenges in Graph Topology Identification

In GTI, the fundamental assumption is that $\mathbf{x} \sim \mathscr{F}(\mathbf{S})$, where $\mathscr{F}$ is a distribution and the relationship between $\mathscr{F}$ and $\mathbf{S}$ is known.

For example $\mathscr{F}$ is a normal distribution, specifically $\mathscr{F} = \mathcal{N}(\boldsymbol{\mu}(\mathbf{S}), \boldsymbol{\Sigma}(\mathbf{S}))$. Here, $\boldsymbol{\mu}(\mathbf{S})$ and $\boldsymbol{\Sigma}(\mathbf{S})$ are the mean vector and covariance matrix, respectively, determined by the structured matrix $\mathbf{S}$. Further, $\boldsymbol{\mu}(\mathbf{S})$ and $\boldsymbol{\Sigma}(\mathbf{S})$ are determined once $\mathbf{S}$ is known.

The main challenge here is how to estimate $\mathbf{S}$ from $\mathbf{X}$ and the prior knowledge of $\mathscr{F}$. This actually encompasses two underlying questions:

1. **Identifiability:** Whether there exists another $\mathbf{S}'$, which could generate $\mathbf{X}$ with the same or higher probability than $\mathbf{S}$? This question addresses whether $\mathbf{S}$ can be uniquely determined from $\mathbf{X}$.
2. **Algorithmic Feasibility:** Is there an algorithm capable of efficiently estimating $\mathbf{S}$ from $\mathbf{X}$? This focuses on the practical aspects of finding a solution.

2.4 Structural Equation Model

We select the structural equation model (SEM) [5] as the starting point for our studies. It excellently supports linear modeling of graph phenomena and is backed by numerous existing studies such as those in [19] and [20].

The SEM provides a versatile statistical modeling framework employed to analyze multivariate datasets across various disciplines, including brain science [4] and genetics [21]. The utility of the SEM lies in its capacity to model complex relationships between observed variables, accounting for both direct and indirect influences, as well as the effects of latent variables.

2.4.1 Mathematical Representation

Given an observation vector $\mathbf{y} \in \mathbb{R}^N$, the SEM establishes a model where each variable y_i in the vector is expressed as a linear combination of other variables in the dataset. This relationship can be formalized as follows:

$$y_i = \sum_{j \neq i} S_{ij} y_j + F_{ii} u_i + e_i, \tag{2.12}$$

where:

- S_{ij} represents the weight of the influence exerted by node j on node i;
- F_{ii} denotes the impact of the exogenous variable u_i on node i;
- e_i captures the residual errors or unmodeled dynamics.

It is important to note that in the SEM, we have replaced the previous notations $\mathbf{x}$ and $\mathbf{X}$ with $\mathbf{y}$ and $\mathbf{Y}$, respectively. This change is primarily to conform to the conventional notation typically used in a SEM.

We can simplify this to a matrix-vector form as follows

$$\mathbf{y} = \mathbf{S}\mathbf{y} + \mathbf{F}\mathbf{u} + \mathbf{e}. \tag{2.13}$$

In case we consider multiple independent realizations of $\mathbf{e}$, which can be stacked in $\mathbf{E} = [\mathbf{e}_1, \mathbf{e}_2, \ldots, \mathbf{e}_T]$, and multiple independent realizations of control inputs $\mathbf{u}$, stacked in $\mathbf{U} = [\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_T]$, we obtain multiple independent realizations of $\mathbf{y}$, grouped in $\mathbf{Y} = [\mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_T]$, as

$$\mathbf{Y} = \mathbf{S}\mathbf{Y} + \mathbf{F}\mathbf{U} + \mathbf{E}. \tag{2.14}$$

The research then bifurcates into two directions, primarily differentiated by whether the exogenous variables are considered.

Case 1. Assuming exogenous variables, we obtain

$$\mathbf{y} = \mathbf{S}\mathbf{y} + \mathbf{F}\mathbf{u} + \mathbf{e}, \tag{2.15}$$

where $\mathbf{S}$ is the matrix whose (i, j)th entry is S_{ij} if $i \neq j$ and 0 otherwise (assuming no self-influence) and $\mathbf{F}$ is the matrix whose ith diagonal entry is F_{ii} and 0 otherwise.

Case 2. If we consider scenarios where the impact of exogenous variables can be ignored, this can be reduced to

$$\mathbf{y} = \mathbf{S}\mathbf{y} + \mathbf{e}, \tag{2.16}$$

In the field of GTI, when focusing on (2.15), researchers are interested in estimating matrices $\mathbf{F}$ and $\mathbf{S}$ through known $\mathbf{y}$ and $\mathbf{u}$. In the study of (2.16), the aim is to estimate $\mathbf{S}$ using $\mathbf{y}$.

Undoubtedly, Case 2 can be considered a special case of Case 1 (when $\mathbf{u} = 0$). This makes Case 1 appear to be a more challenging problem. However, it is not really true because in Case 1, it is usually assumed that endogenous factors are predominant and known, meaning that $\mathbf{e}$ is very small relative to $\mathbf{F}\mathbf{u}$ and $\mathbf{u}$ is known. In contrast, in the study of Case 2, $\mathbf{e}$ is considered the main factor, in which case, $\mathbf{e}$ can indeed be seen as representing some unmodeled dynamics.

Despite extensive research on the SEM prior to its application in graph theory, and the apparent simplicity of the SEM as a linear model, its implementation is not as straightforward as expected. As shown in [22] and [11], approaches to tackle Case 1, require observations of exogenous variables and assumptions about sparsity.

In the context of Case 2, a remarkable result was provided by [8], where it was demonstrated that for sparse directed acyclic graphs (DAGs), the graph can be uniquely determined when the unknown external inputs are Gaussian with equal variance. This foundational work spurred further developments, leading to more efficient algorithms as evidenced by [9] and [10]. However, for undirected graphs, methods using signal matching have performed poorly without the presence of exogenous variables, even with the introduction of sparsity constraints [11, 12]. On the other hand, spectral template-based approaches, such as the polynomial graphical lasso (PGL) algorithm, have shown potential in handling certain graphs [2], but the results typically differ from the true structure by a scale factor and require extensive restrictions (sparsity and sign) on the graph for the method to be effective.

In this thesis, we primarily investigate the scenario described in Case 2, proposing solutions for both undirected and directed graphs (sparse but potentially containing cycles).

The detailed discussions of these solutions will be presented in the subsequent two chapters. But first, we will review existing literature on Case 1 and Case 2 separately, which will provide a solid foundation and a clear starting point for our problem.

2.4.2 Identifiability of Case 1

We begin our discussion from a relatively simple perspective, where both the process and the conclusions are elegant, and there is even a closed-form solution available.

In [7] we can find an interesting theorem for Case 1.

Theorem 1. Suppose that data $\mathbf{U}$ and $\mathbf{Y}$ abide to the SEM $\mathbf{Y} = \mathbf{SY} + \mathbf{FU}$, for a matrix $\mathbf{S}$ with diagonal entries $S_{ii} = 0$ and diagonal matrix $\mathbf{F}$ with diagonal entries $F_{ii} \neq 0$. If $\mathbf{U}$ has full row rank, then $\mathbf{S}$ and $\mathbf{F}$ are uniquely expressible in terms of $\mathbf{U}$ and $\mathbf{Y}$ as $\mathbf{F} = \text{Diag}^{-1}((\mathbf{YU}^{\dagger})^{-1})$ and $\mathbf{S} = \mathbf{I} - \mathbf{F}\left(\mathbf{YU}^{\dagger}\right)^{-1}$ with $\text{Diag}(\cdot)$ setting the off-diagonal elements to zero and $(\cdot)^{\dagger}$ taking the pseudo-inverse.

This theorem can be viewed as a result for the ideal case where $\mathbf{e} = \mathbf{0}$ strictly holds. Then, provided that $\mathbf{U}$ is sufficiently exciting, $\mathbf{S}^*$ can be uniquely determined. While the original paper's derivation is algebraic, here, to integrate it with the other sections, we want to achieve the same results through an optimization approach. In other words, we will demonstrate that $\mathbf{F}^* = \text{Diag}^{-1}((\mathbf{YU}^{\dagger})^{-1})$ and $\mathbf{S}^* = \mathbf{I} - \mathbf{F}^*\left(\mathbf{YU}^{\dagger}\right)^{-1}$ are the unique solutions to the following optimization problem (under the assumptions of the theorem):

$$\begin{aligned} \hat{\mathbf{F}}^*, \hat{\mathbf{S}}^* = \underset{\hat{\mathbf{F}},\hat{\mathbf{S}}}{\arg\min} \quad & \|\mathbf{Y} - \hat{\mathbf{S}}\mathbf{Y} - \hat{\mathbf{F}}\mathbf{U}\|_F^2 \\ \text{subject to} \quad & \hat{S}_{ii} = 0, \\ & \hat{F}_{ij} = 0, \quad i \neq j. \end{aligned} \tag{2.17}$$

Let

$$f(\hat{\mathbf{S}}, \hat{\mathbf{F}}) = \|\mathbf{Y} - \hat{\mathbf{S}}\mathbf{Y} - \hat{\mathbf{F}}\mathbf{U}\|_F^2 \tag{2.18}$$

which can be expressed in trace form as:

$$f(\hat{\mathbf{S}}, \hat{\mathbf{F}}) = \text{tr}((\mathbf{Y} - \hat{\mathbf{S}}\mathbf{Y} - \hat{\mathbf{F}}\mathbf{U})^{\top}(\mathbf{Y} - \hat{\mathbf{S}}\mathbf{Y} - \hat{\mathbf{F}}\mathbf{U})) \tag{2.19}$$

The derivative of f with respect to $\hat{\mathbf{F}}$ is given by:

$$\frac{\partial f}{\partial \hat{\mathbf{F}}} = 2\mathbf{U}\mathbf{U}^{\top}\hat{\mathbf{F}} - 2(\mathbf{I} - \hat{\mathbf{S}})\mathbf{Y}\mathbf{U}^{\top} \tag{2.20}$$

We now prove setting this derivative to zero yields a unique solution $\mathbf{F}^*$, as $\mathbf{U}$ has full row rank. This solution can be expressed as:

$$\mathbf{F}^* = (\mathbf{I} - \mathbf{S}^*)\mathbf{Y}\mathbf{U}^{\top}(\mathbf{U}\mathbf{U}^{\top})^{-1} = (\mathbf{I} - \mathbf{S}^*)\mathbf{Y}\mathbf{U}^{\dagger} \tag{2.21}$$

Here, the matrix $\mathbf{U}\mathbf{U}^\top$ is invertible because $\mathbf{U}$ has full row rank. From the assumption, it is known that $(\mathbf{I}-\mathbf{S}^*)\mathbf{Y} = \mathbf{F}^*\mathbf{U}$ and that $\mathbf{F}^*\mathbf{U}$ is full row rank. Consequently, $(\mathbf{I}-\mathbf{S}^*)\mathbf{Y}$ is full row rank and thus $\mathbf{I}-\mathbf{S}^*$ is invertible. Since $\mathbf{F}^*$ is invertible, and considering that $\mathbf{I}-\mathbf{S}^*$ is invertible, it follows that $\mathbf{Y}\mathbf{U}^\dagger$ must also be invertible.

By inverting $\mathbf{F}^*$, we then obtain:

$$(\mathbf{F}^*)^{-1} = (\mathbf{Y}\mathbf{U}^\dagger)^{-1}(\mathbf{I}-\mathbf{S}^*)^{-1} \tag{2.22}$$

and thus

$$(\mathbf{F}^*)^{-1}(\mathbf{I}-\mathbf{S}^*) = (\mathbf{Y}\mathbf{U}^\dagger)^{-1} \tag{2.23}$$

Since $\mathbf{F}^*$ is a diagonal matrix, $(\mathbf{F}^*)^{-1}$ is also a diagonal matrix, and therefore we have

$$\mathrm{diag}((\mathbf{F}^*)^{-1}(\mathbf{I}-\mathbf{S}^*)) = \mathrm{diag}((\mathbf{F}^*)^{-1}) = \mathrm{diag}((\mathbf{Y}\mathbf{U}^\dagger)^{-1}) \tag{2.24}$$

This leads to:

$$(\mathbf{F}^*)^{-1} = \mathrm{diag}((\mathbf{Y}\mathbf{U}^\dagger)^{-1}) \tag{2.25}$$

Hence, we have demonstrated that $\mathbf{F}^*$ is defined as:

$$\mathbf{F}^* = \mathrm{diag}((\mathbf{Y}\mathbf{U}^\dagger)^{-1})^{-1} \tag{2.26}$$

Once $\mathbf{F}^*$ has been determined, solving for $\mathbf{S}^*$ becomes straightforward. We omit the proof here for brevity.

It is meaningful to formulate the original problem as (2.17) because this allows us to incorporate additional sparsity constraints, such as L1 regularization $\|\mathbf{S}\|_1$. This enables us to obtain a more sparse structure, which can yield better results compared to merely calculating $\mathbf{F}^* = \mathrm{Diag}^{-1}((\mathbf{Y}\mathbf{U}^\dagger)^{-1})$ and $\mathbf{S}^* = \mathbf{I} - \mathbf{F}^*\left(\mathbf{Y}\mathbf{U}^\dagger\right)^{-1}$, especially when $\mathbf{e} \neq \mathbf{0}$.

2.4.3 Identifiability of Case 2

In the context of Case 2, we first present a theorem that proves identifiability under certain conditions. Subsequently, we aim to explore this theorem to illustrate its connection with problem (2.17). We aim to use the connection between the two theorems as a promising starting point to explain where our work begins.

Theorem 2. In [8] it is proven that the matrix $\mathbf{S}$ is identifiable if it represents a directed acyclic graph (DAG) and the error term $\mathbf{e}$ follows a normal distribution $\mathcal{N}(0, \sigma^2\mathbf{I})$.

Although this theorem appears straightforward, the proof is quite complex, requiring numerous lemmas.

This theorem indeed addresses the first issue, indicating that different DAGs $\mathcal{G}$ and $\mathcal{G}'$ will lead to different results for $\mathbf{S}$. Here, I merely wish to present an outline of what their optimization problem establishes. This actually represents another viewpoint of their theorem, namely that the optimization problem they propose is capable of converging to the correct result.

Using (2.16), we know that $\mathbf{y} = (\mathbf{I} - \mathbf{S})^{-1}\mathbf{e}$, so $\mathbf{y}$ is distributed as $\mathcal{N}(0, (\mathbf{I} - \mathbf{S})^{-1}(\mathbf{I} - \mathbf{S})^{-\top})$. Then, the probability density function $p(\mathbf{y})$ for $\mathbf{y}$ is given by:

$$p(\mathbf{y}) = \frac{1}{(2\pi)^{N/2}|\boldsymbol{\Sigma}_{\mathbf{y}}|^{1/2}} \exp\left(-\frac{1}{2}\mathbf{y}^\top \boldsymbol{\Sigma}_{\mathbf{y}}^{-1}\mathbf{y}\right) \tag{2.27}$$

where $\boldsymbol{\Sigma}_{\mathbf{y}} = (\mathbf{I} - \mathbf{S})^{-1}(\mathbf{I} - \mathbf{S})^{-\top}$ is the covariance matrix.

Let $f = -\log p(\mathbf{Y})$, then

$$f = \frac{NT}{2}\log\left(2\pi\sigma^2\right) + \frac{1}{2\sigma^2}\operatorname{tr}\left\{\mathbf{Y}^\top(\mathbf{I} - \mathbf{S})^\top(\mathbf{I} - \mathbf{S})\mathbf{Y}\right\} \tag{2.28}$$

Applying the trace property, it can also be written as

$$f = \frac{NT}{2}\log\left(2\pi\sigma^2\right) + \frac{T}{2\sigma^2}\operatorname{tr}\left\{(\mathbf{I} - \mathbf{S})^\top(\mathbf{I} - \mathbf{S})\mathbf{C}_{\mathbf{y}}\right\} \tag{2.29}$$

where $\mathbf{C}_{\mathbf{y}} = \frac{1}{T}\mathbf{Y}\mathbf{Y}^\top$.

Define the set of all adjacency matrices of directed acyclic graphs as $\mathscr{S}$. Then it can be proven that the solution to

$$\sigma^{2*}, \mathbf{S}^* = \operatorname*{argmin}_{\hat{\sigma}^2>0, \hat{\mathbf{S}}\in\mathscr{S}} f(\hat{\sigma}^2, \hat{\mathbf{S}}) + \lambda\|\hat{\mathbf{S}}\|_0 \tag{2.30}$$

converges to the true values of σ^2 and $\mathbf{S}$ as T approaches infinity, where λ is a scale value only related to the sample number.

Although the theorem is powerful, the inclusion of the L_0 norm often leads to optimization difficulties. Consequently, several studies [9, 10] have transformed this problem by approximating the L_0 norm with the L_1 norm. This approximation yields very effective results.

2.4.4 Connecting the Methods

Now, we would like to discuss some connections between (2.28) and (2.17). In (2.28), if we assume that σ is known and we ignore the sparsity, then the problem is solely related to minimizing the expression $\operatorname{tr}\left\{\mathbf{Y}^\top(\mathbf{I} - \mathbf{S})^\top(\mathbf{I} - \mathbf{S})\mathbf{Y}\right\}$. From (2.19), it is evident that when $\mathbf{U} = \mathbf{0}$, then also (2.17) reduces to minimizing $\operatorname{tr}\left\{\mathbf{Y}^\top(\mathbf{I} - \mathbf{S})^\top(\mathbf{I} - \mathbf{S})\mathbf{Y}\right\}$.

Therefore, the method of [7] can be viewed as minimizing the difference between $\mathbf{Y}$ and $\mathbf{SY} + \mathbf{FU}$. Similarly, the approach of [8] can be considered as minimizing the difference between $\mathbf{Y}$ and $\mathbf{SY} + \mathbf{FU}$ under known σ^2.

Based on this foundational understanding, we will propose a signal matching method in the following chapter, which is detailed in Section 3.1.

2.5 Binary Quadratic Programming

The motivation for introducing binary quadratic programming (BQP) stems from its relevance in the subsequent derivations concerning estimating an undirected SEM. In the next chapter, we will transform this GTI problem into a BQP format.

BQP involves minimizing a quadratic function subject to binary constraints on the variables. It is widely recognized for its computational challenges, especially due to its NP-hard nature. In this context, we utilize an approximation technique called semi-definite relaxation (SDR) [23]. This method approaches the solution by relaxing the original optimization problem into a convex form. Although this relaxation is an approximation, it is powerful and effective for handling NP-hard problems. For a detailed discussion on the efficacy and theoretical underpinnings of SDR, see [24].

2.5.1 Problem Formulation

Let us first describe the basic form of BQP [25]. Suppose $\mathbf{C}$ is a positive semi-definite matrix, the variables $\mathbf{x} \in \{-1, 1\}^N$, and the objective is to minimize $\mathbf{x}^\top \mathbf{C}\mathbf{x}$. The problem can then be stated as:

$$\min_{\mathbf{x} \in \{-1,1\}^N} \mathbf{x}^\top \mathbf{C}\mathbf{x} \tag{2.31}$$

2.5.2 Semi-Definite Relaxation

A powerful relaxation technique for this problem is described in [25], known as SDR. By defining $\mathbf{X} = \mathbf{x}\mathbf{x}^\top$, the objective function can be reformulated in terms of the trace function, transforming the quadratic form into a linear form over the matrix variable $\mathbf{X}$ we obtain:

$$\mathbf{x}^\top \mathbf{C}\mathbf{x} = \operatorname{tr}(\mathbf{C}\mathbf{x}\mathbf{x}^\top) = \operatorname{tr}(\mathbf{C}\mathbf{X}) \tag{2.32}$$

To maintain $\mathbf{X} = \mathbf{x}\mathbf{x}^\top$ and $\mathbf{x} \in \{-1, 1\}^N$, it is equivalent to having $\operatorname{rank}(\mathbf{X}) = 1$, $\operatorname{diag}(\mathbf{X}) = \mathbf{1}$, and $\mathbf{X} \succeq 0$. It is straightforward to derive that if $\mathbf{X} = \mathbf{x}\mathbf{x}^\top$ and $\mathbf{x} \in \{-1, 1\}^N$, then $\operatorname{rank}(\mathbf{X}) = 1$, $\operatorname{diag}(\mathbf{X}) = \mathbf{1}$, and $\mathbf{X} \succeq 0$. To prove the converse, assume $\operatorname{rank}(\mathbf{X}) = 1$, $\operatorname{diag}(\mathbf{X}) = \mathbf{1}$, and $\mathbf{X} \succeq 0$. Since $\operatorname{rank}(\mathbf{X}) = 1$ and $\mathbf{X} \succeq 0$, $\mathbf{X}$ can be written as $\mathbf{X} = \mathbf{x}\mathbf{x}^\top$. The condition $\operatorname{diag}(\mathbf{X}) = \mathbf{1}$ ensures $x_i^2 = 1$ for all i, and thus $\mathbf{x} \in \{-1, 1\}^N$.

Thus, solving the BQP problem is equivalent to solving the following problem:

$$\begin{aligned} \min_{\mathbf{X}} \quad & \operatorname{tr}(\mathbf{C}\mathbf{X}) \\ \text{subject to} \quad & \mathbf{X} \succeq \mathbf{0}, \\ & \operatorname{diag}(\mathbf{X}) = \mathbf{1}, \\ & \operatorname{rank}(\mathbf{X}) = 1 \end{aligned} \tag{2.33}$$

The relaxed problem, removing the rank constraint, is expressed as follows:

$$\begin{aligned} \min_{\mathbf{X}} \quad & \operatorname{tr}(\mathbf{C}\mathbf{X}) \\ \text{subject to} \quad & \mathbf{X} \succeq \mathbf{0}, \\ & \operatorname{diag}(\mathbf{X}) = \mathbf{1} \end{aligned} \tag{2.34}$$

This problem is a convex optimization problem, which is classified as a semi-definite programming (SDP) problem.

Another approach to solve the BQP problem is the branch and bound method. This technique is supported by the cvxpy solver GUROBI [26], which accommodates 0-1 programming. By making simple adjustments to the code, we can directly apply this method for BQP. However, despite its applicability, the branch and bound method has several disadvantages. First, it can be computationally expensive, especially for larger problems where the number of possible solutions grows exponentially. This growth significantly increases the computational time, making it impractical for large-scale applications. Additionally, while this method guarantees finding the global optimum, the time required to reach convergence can be prohibitive, limiting its utility in scenarios demanding quick results.

2.5.3 Extension to General BQP

We now consider extending the original optimization problem which minimizes $\mathbf{x}^\top \mathbf{C}\mathbf{x}$ by including a linear term and a constant, which allows the problem to take the form:

$$\min_{\mathbf{x} \in \{-1,1\}^N} \mathbf{x}^\top \mathbf{C}\mathbf{x} + \mathbf{c}^\top \mathbf{x} + c \tag{2.35}$$

To solve this, we introduce an augmented vector $\mathbf{x}' = [1, \mathbf{x}^\top]^\top$. Defining $\mathbf{X}' = \mathbf{x}'\mathbf{x}'^\top$, the extended objective function can then be equivalently expressed as $\operatorname{tr}(\mathbf{C}'\mathbf{X}')$, where the augmented matrix $\mathbf{C}'$ is defined as:

$$\mathbf{C}' = \begin{pmatrix} c & \frac{1}{2}\mathbf{c}^\top \\ \frac{1}{2}\mathbf{c} & \mathbf{C} \end{pmatrix} \tag{2.36}$$

Since $\mathbf{X}'$ is still a positive semi-definite matrix and its diagonal is $\mathbf{1}$, we finally obtain the problem

$$\begin{aligned} \min \quad & \operatorname{Tr}(\mathbf{C}'\mathbf{X}') \\ \text{s.t.} \quad & \mathbf{X}' \succeq \mathbf{0}, \\ & \operatorname{diag}(\mathbf{X}') = \mathbf{1}, \\ & \operatorname{rank}(\mathbf{X}') = 1 \end{aligned} \tag{2.37}$$

Again, dropping the rank constraint, we obtain the following SDP problem:

$$\begin{aligned} \min \quad & \operatorname{Tr}(\mathbf{C}'\mathbf{X}') \\ \text{s.t.} \quad & \mathbf{X}' \succeq \mathbf{0}, \\ & \operatorname{diag}(\mathbf{X}') = \mathbf{1}. \end{aligned} \tag{2.38}$$

2.6 Discrete Quadratic Programming

Discrete quadratic programming (DQP) is an extension of binary quadratic programming (BQP). Essentially, it deals with optimization problems that extend (2.31) by expanding the variable range from $\{-1, 1\}$ to any discrete interval. That is

$$\begin{aligned} \min_{\mathbf{x}} \quad & \mathbf{x}^\top \mathbf{C} \mathbf{x} \\ \text{subject to} \quad & \mathbf{x} \in D_1 \times \cdots \times D_n, \end{aligned} \tag{2.39}$$

where each $D_i = \{c_i^1, c_i^2, \ldots, c_i^{p_i}\}$ is a finite subset of $\mathbb{R}$, and every c_i^j is a candidate for D_i.

Here, we will explain an SDR trick [27] to solve this problem. This trick is basically an extension of what we used to solve the BQP.

For the objective, let

$$\tilde{\mathbf{C}} = \begin{pmatrix} 0 & \mathbf{0}^\top \\ \mathbf{0} & \mathbf{C} \end{pmatrix} \tag{2.40}$$

and define

$$\mathbf{x}' = \begin{bmatrix} 1 \\ \mathbf{x} \end{bmatrix} \tag{2.41}$$

Following the idea of SDR, we introduce

$$\mathbf{X}' = \mathbf{x}'\mathbf{x}'^T \tag{2.42}$$

Let our indexing of $\mathbf{X}'$ start from 0, and let X'_{ij} represent the element at the i-th row and j-th column of $\mathbf{X}'$. Then we have:

$$x_i^2 = X'_{ii}, \quad x_i = X'_{0i} \tag{2.43}$$

For $x_i \in \{c_i^1, c_i^2, \ldots, c_i^{p_i}\}$, the value range of (x_i, x_i^2) can be relaxed to a convex hull formed by the corner points $(c_i^j, (c_i^j)^2)$.

To illustrate this convex hull, let's take an example. Consider plotting the result for c_i^j on the curve $y = x^2$. Suppose we have four points: c_i^1, c_i^2, c_i^3, and c_i^4. On the graph of $y = x^2$, these points correspond to the blue dots representing

$$\{(c_i^1, (c_i^1)^2), (c_i^2, (c_i^2)^2), (c_i^3, (c_i^3)^2), (c_i^4, (c_i^4)^2)\} \tag{2.44}$$

By determining the convex hull of these four points, we obtain the feasible region for (X'_{0i}, X'_{ii}). This region is illustrated in purple in Fig. 2.1.

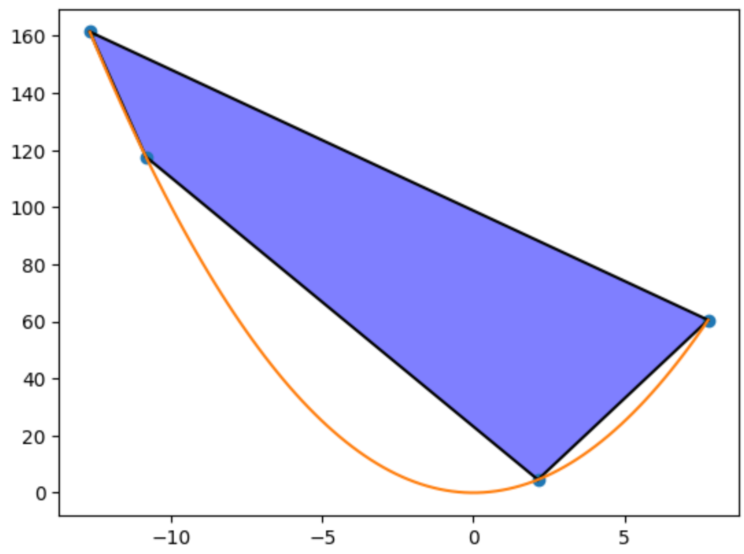


Figure 2.1: Convex hull formed by four blue points on $y = x^2$

Denoting the convex hull for index i as $\mathcal{CH}_i$, our problem simplifies to

$$\begin{aligned} \min_{\mathbf{X}'} \quad & \operatorname{tr}(\tilde{\mathbf{C}}\mathbf{X}') \\ \text{subject to} \quad & (X'_{0i}, X'_{ii}) \in \mathcal{CH}_i \text{ for } i = 1, 2, \ldots, N, \\ & \mathbf{X}' \succeq 0, \\ & X'_{00} = 1. \end{aligned} \tag{2.45}$$

This approach evidently encompasses the BQP, as for BQP, each $D_i = \{-1, 1\}$. The condition $(X'_{0i}, X'_{ii}) \in \mathcal{CH}_i$ enforces $X'_{ii} = 1$, which is the same as for a BQP.

Extending GTI for the SEM 3

In this project, we primarily focus on extending Case 2, which states that $\mathbf{y} = \mathbf{S}\mathbf{y} + \mathbf{e}$. Considering theorem 2, there are two constraints: first, that $\mathbf{S}$ must represent a directed acyclic graph (DAG), and second, that $\mathbf{e}$ must follow a Gaussian distribution with equal variance. With respect to these constraints, we propose two directions for extension: first, by generalizing $\mathbf{S}$ to another class of graphs; second, by broadening $\mathbf{e}$ to more general distributions.

In this chapter, we first consider undirected graphs and initially experiment with a trivial method referred to as signal matching (SigMatch). However, we will prove that this approach fails to consistently converge to the correct graph structure. Subsequently, we shift our focus to the concept of covariance matching (CovMatch), deriving an optimization problem that consistently converges to the correct solution under relatively lenient conditions. Furthermore, we extend our consideration of $\mathbf{e}$ to any known Gaussian distribution. Finally, we provide experiments validating the effectiveness of our method and compare it with other approaches.

We begin with the extension of $\mathbf{S}$ to a class of graphs beyond the class of DAGs. For ease of discussion, we assume that $\mathbf{e} \sim \mathcal{N}(0, \mathbf{I})$. Given that $\mathbf{y} = (\mathbf{I} - \mathbf{S})^{-1}\mathbf{e}$, this leads to the mean of $\mathbf{y}$, denoted $\boldsymbol{\mu}_{\mathbf{y}}$, being zero. And the covariance matrix $\boldsymbol{\Sigma}_{\mathbf{y}}$ can be derived as:

$$\boldsymbol{\Sigma}_{\mathbf{y}} = \mathbb{E}[\mathbf{y}\mathbf{y}^\top] = (\mathbf{I} - \mathbf{S})^{-1}\mathbb{E}[\mathbf{e}\mathbf{e}^\top](\mathbf{I} - \mathbf{S})^{-\top} = (\mathbf{I} - \mathbf{S})^{-1}(\mathbf{I} - \mathbf{S})^{-\top}. \tag{3.1}$$

Corresponding to the discussions in Subsection 2.3.1, this formulation implies that $\mathbf{y}$ follows a normal distribution, $\mathbf{y} \sim \mathcal{N}(\boldsymbol{\mu}(\mathbf{S}), \boldsymbol{\Sigma}(\mathbf{S}))$, where $\boldsymbol{\mu}(\mathbf{S}) = \mathbf{0}$ and $\boldsymbol{\Sigma}(\mathbf{S}) = (\mathbf{I} - \mathbf{S})^{-1}(\mathbf{I} - \mathbf{S})^{-\top}$. If further $\mathbf{S}$ is symmetric, then the covariance simplifies to $\boldsymbol{\Sigma}(\mathbf{S}) = (\mathbf{I} - \mathbf{S})^{-2}$.

3.1 Signal Matching

3.1.1 Problem Formulation

A straightforward idea is to extend $\mathbf{S}$ to symmetric graphs without self-loops. To distinguish between the estimated and the actual $\mathbf{S}$, we use $\hat{\mathbf{S}}$ for the estimated graph. It is well known that symmetry corresponds to $\hat{\mathbf{S}} = \hat{\mathbf{S}}^\top$, and the absence of self-loops is represented by $\text{diag}(\hat{\mathbf{S}}) = \mathbf{0}$.

Following our previous discussion, we aim to minimize the difference between $\mathbf{Y}$ and $\mathbf{S}\mathbf{Y} + \mathbf{F}\mathbf{U}$, where $\mathbf{U} = \mathbf{0}$. Thus, we naturally obtain the following optimization problem:

Optimization Problem 1.

$$\begin{aligned} \min_{\hat{\mathbf{S}}} \quad & \frac{1}{T}\|\mathbf{Y} - \hat{\mathbf{S}}\mathbf{Y}\|_F^2 \\ \text{subject to} \quad & \hat{\mathbf{S}} = \hat{\mathbf{S}}^T, \\ & \operatorname{diag}(\hat{\mathbf{S}}) = \mathbf{0}. \end{aligned} \tag{3.2}$$

In trace form this is equal to

$$\begin{aligned} \min_{\hat{\mathbf{S}}} \quad & \operatorname{tr}\left(\hat{\mathbf{S}}^2\mathbf{C}_{\mathbf{y}}\right) - 2\operatorname{tr}(\hat{\mathbf{S}}\mathbf{C}_{\mathbf{y}}) + \operatorname{tr}(\mathbf{C}_{\mathbf{y}}) \\ \text{subject to} \quad & \hat{\mathbf{S}} = \hat{\mathbf{S}}^T, \\ & \operatorname{diag}(\hat{\mathbf{S}}) = \mathbf{0}, \end{aligned} \tag{3.3}$$

where $\mathbf{C}_{\mathbf{y}} = \frac{1}{T}\mathbf{Y}\mathbf{Y}^T$ represents the sample covariance matrix.

3.1.2 Identifiability Analysis

The formulation of Optimization Problem 1 is not new, see for instance [12]. However, in our testing, although this method can produce approximately correct graphs for simple structures, it generally fails to correctly recover the graph in more complex scenarios. This raises a significant misconception: it is unclear whether the method itself is inadequate, or if the failure to converge to the correct result is due to insufficient observations. To address this, we consider a hypothetical scenario with an infinite number of observations, or equivalently, we consider the expected value of the objective function. In other words, we consider a new optimization problem:

Optimization Problem 2.

$$\begin{aligned} \mathbf{S}^* = \arg\min_{\hat{\mathbf{S}}} \quad & \mathbb{E}\{\|\mathbf{y} - \hat{\mathbf{S}}\mathbf{y}\|^2\} \\ \text{subject to} \quad & \hat{\mathbf{S}} = \hat{\mathbf{S}}^\top \\ & \operatorname{diag}(\hat{\mathbf{S}}) = 0 \end{aligned} \tag{3.4}$$

By doing that, we can successfully prove that it is impossible to correctly reconstruct the graph solely through the Optimization Problem 2. This also indicates that for simple graphs, the earlier mentioned results were only somewhat related to the true graph, they were not identical. This lack of identifiability is shown in the next theorem.

Theorem 3. Assume that the signal and the graph satisfy the SEM relation $\mathbf{y} = \mathbf{S}\mathbf{y}+\mathbf{e}$, where $\mathbf{e} \sim \mathcal{N}(0, \mathbf{I})$, and $\mathbf{S}$ is a non-zero hollow symmetric matrix. Then it follows that the solution to Optimization Problem 2 is not exact, i.e., $\mathbf{S}^* \neq \mathbf{S}$.

Proof. In this proof, we will make use of the following two lemmas.

Lemma 1. If matrix $\mathbf{A}$ is symmetric, then $\mathbf{A}$ is diagonalizable and its eigenvectors can form an orthogonal matrix. In other words, there exists an orthogonal matrix $\mathbf{P}$ and a diagonal matrix $\mathbf{D}$ such that $\mathbf{A} = \mathbf{P}\mathbf{D}\mathbf{P}^\top$, where $\mathbf{P}^\top = \mathbf{P}^{-1}$, demonstrating that $\mathbf{P}$ is an orthogonal matrix.

Lemma 2. If matrices $\mathbf{A}$ and $\mathbf{B}$ are diagonalizable, such that $\mathbf{A} = \mathbf{P_A D_A P_A^{-1}}$ and $\mathbf{B} = \mathbf{P_B D_B P_B^{-1}}$, then $\mathbf{AB} = \mathbf{BA}$ if and only if $\mathbf{A}$ and $\mathbf{B}$ are simultaneously diagonalizable [28].

We start from the basic expression of the error:

$$\begin{aligned} &\|\mathbf{y} - \hat{\mathbf{S}}\mathbf{y}\|^2 \\ =&\|(\mathbf{I} - \hat{\mathbf{S}})\mathbf{y}\|^2 \\ =&\mathbf{y}^\top(\mathbf{I} - \hat{\mathbf{S}})^\top(\mathbf{I} - \hat{\mathbf{S}})\mathbf{y} \\ =&\operatorname{tr}\left((\mathbf{I} - \hat{\mathbf{S}})^2(\mathbf{y}\mathbf{y}^\top)\right) \end{aligned} \tag{3.5}$$

Since $\mathbb{E}\{\mathbf{y}\mathbf{y}^\top\} = \mathbb{E}\{(\mathbf{I} - \mathbf{S})^{-\mathbf{1}}\mathbf{e}\mathbf{e}^\top(\mathbf{I} - \mathbf{S})^{-\top}\} = (\mathbf{I} - \mathbf{S})^{-2}$, the objective becomes

$$f(\hat{\mathbf{S}}) = \operatorname{tr}\left((\mathbf{I} - \mathbf{S})^{-2}(\mathbf{I} - \hat{\mathbf{S}})^2\right) \tag{3.6}$$

Without considering any constraints, we now differentiate with respect to $\hat{\mathbf{S}}$, which leads to

$$\frac{\partial f}{\partial \hat{\mathbf{S}}} = (\mathbf{I} - \mathbf{S})^{-2}\hat{\mathbf{S}} + \hat{\mathbf{S}}(\mathbf{I} - \mathbf{S})^{-2} - 2(\mathbf{I} - \mathbf{S})^{-2} \tag{3.7}$$

If $\hat{\mathbf{S}} = \mathbf{S}$, and $\hat{\mathbf{S}}$ is also the optimal solution, we aim to derive a contradiction.

Considering the symmetry of $\mathbf{S}$, according to [29], the derivative becomes

$$\frac{df}{d\hat{\mathbf{S}}} = \frac{\partial f}{\partial \hat{\mathbf{S}}} + \frac{\partial f}{\partial \hat{\mathbf{S}}^\top} - \operatorname{Diag}\frac{\partial f}{\partial \hat{\mathbf{S}}} \tag{3.8}$$

Since $\frac{\partial f}{\partial \hat{\mathbf{S}}} = \frac{\partial f}{\partial \hat{\mathbf{S}}^\top}$ and $\frac{df}{d\hat{\mathbf{S}}}$ should be a diagnal matrix, then also $\frac{\partial f}{\partial \hat{\mathbf{S}}}$ should be a diagonal matrix.

In conclusion, $\frac{\partial f}{\partial \hat{\mathbf{S}}}$ may be non-zero only on the diagonal, and should be zero elsewhere. That is,

$$\frac{\partial f}{\partial \hat{\mathbf{S}}} = \operatorname{diag}(\boldsymbol{\lambda}),\ \ \boldsymbol{\lambda} \in \mathbb{R}^{N\times 1} \tag{3.9}$$

According to Lemma 1, the matrix $\mathbf{S}$ is diagonalizable. Let the eigenvalue decomposition (EVD) of $\mathbf{S}$ be $\mathbf{U}\boldsymbol{\Lambda}\mathbf{U}^T$, where $\mathbf{U}$ is an orthogonal matrix and $\boldsymbol{\Lambda}$ is a diagonal matrix containing the eigenvalues of $\mathbf{S}$. Consequently, the inverse square of $\mathbf{I} - \mathbf{S}$ can be expressed as:

$$(\mathbf{I} - \mathbf{S})^{-2} = \mathbf{U}(\mathbf{I} - \boldsymbol{\Lambda})^{-2}\mathbf{U}^T \tag{3.10}$$

So $\mathbf{S}$ and $(\mathbf{I} - \mathbf{S})^{-2}$ are simultaneously diagonalizable. Referring to Lemma 2, $\mathbf{S}$ and $(\mathbf{I} - \mathbf{S})^{-2}$ commute, and hence we can write

$$\begin{aligned} \frac{\partial f}{\partial \hat{\mathbf{S}}} &= 2(\mathbf{I} - \mathbf{S})^{-2}\hat{\mathbf{S}} - 2(\mathbf{I} - \mathbf{S})^{-2} \\ &= 2(\mathbf{I} - \mathbf{S})^{-2}(\hat{\mathbf{S}} - \mathbf{I}) \\ &= -2(\mathbf{I} - \mathbf{S})^{-1}, \end{aligned} \tag{3.11}$$

where the last equation is obtained by plugging in as optimal solution $\hat{\mathbf{S}} = \mathbf{S}$.

As we mentioned, optimality means that

$$-2(\mathbf{I} - \mathbf{S})^{-1} = \operatorname{diag}(\boldsymbol{\lambda}),$$

leading to (note that $\mathbf{x} = \mathbf{S}\mathbf{x} + \mathbf{e}$ is indefinite when $\mathbf{I} - \mathbf{S}$ is not invertible)

$$\mathbf{S} = \mathbf{I} - \operatorname{diag}(\boldsymbol{\lambda}/2)^{-1}$$

Thus, $\mathbf{S}$ should be a diagonal matrix which leads to a contradiction.

3.2 Covariance Matching

In this section, we introduce the first successful method capable of recovering a symmetric GSO $\mathbf{S}$ from a SEM. We begin by discussing the motivation behind this approach. We then proceed to derive the related optimization problem, specifically a BQP problem.

3.2.1 Motivation

The following exposition encapsulates the thought process underlying our approach to this problem.

Initially, we consider the relationship $\mathbf{y} = (\mathbf{I} - \mathbf{S})^{-1}\mathbf{e}$, where $\mathbf{e}$ follows a zero mean white Gaussian distribution, thereby implying that $\mathbf{y}$ is also Gaussian. For Gaussian distributions, knowing the mean and the covariance matrix is tantamount to possessing complete knowledge of the distribution. For the considered SEM, the mean zero, and the covariance matrix is $(\mathbf{I} - \mathbf{S})^{-2}$.

We typically approximate the true covariance matrix $\boldsymbol{\Sigma}_{\mathbf{y}}$ with the observed covariance matrix $\mathbf{C}_{\mathbf{y}} = \frac{\mathbf{Y}\mathbf{Y}^\top}{T}$. Under ideal circumstances, as the number of observations tends toward infinity, we have:

$$\lim_{T \to \infty} \mathbf{C}_{\mathbf{y}} = \boldsymbol{\Sigma}_{\mathbf{y}}$$

and this estimation becomes our best possible approximation of the covariance matrix. This understanding forms the starting point of our approach: beginning with the observed sample covariance matrix, which essentially retains all pertinent information about the distribution, we try to fit this to the theoretical covariance matrix which solely depends on the GSO.

3.2.2 Covariance Matching Optimization Problem

Let us again start from

$$\mathbf{y} = (\mathbf{I} - \mathbf{S})^{-1}\mathbf{e}$$

Denoting $(\mathbf{I} - \mathbf{S})^{-1}$ as $\mathbf{H}$, we then have

$$\mathbf{y} \sim \mathcal{N}(\mathbf{0}, \mathbf{H}\mathbf{H}^\top)$$

Note that since $\mathbf{S}$ is symmetric, $\mathbf{H}$ is also symmetric.

Now we basically want to estimate $\mathbf{H}$. To distinguish with the true $\mathbf{H}$, let us assume the estimated variable is $\hat{\mathbf{H}}$ and $\hat{\mathbf{H}} = (\mathbf{I} - \hat{\mathbf{S}})^{-2}$. Following our covariance matching idea, our objective is to estimate $\mathbf{H}$ such that it reproduces the same sample covariance. Assuming that $\mathbf{C}_{\mathbf{y}} = \frac{\mathbf{Y}\mathbf{Y}^\top}{T}$, we formulate the following optimization problem

$$\begin{aligned} \mathbf{H}^* = \arg\min_{\hat{\mathbf{H}}} \quad & \|\hat{\mathbf{H}}\hat{\mathbf{H}}^\top - \mathbf{C}_{\mathbf{y}}\|_F \\ \text{subject to} \quad & \operatorname{diag}(\hat{\mathbf{H}}^{-1}) = \mathbf{1}, \\ & \hat{\mathbf{H}} = \hat{\mathbf{H}}^\top \end{aligned} \tag{3.12}$$

This question is hard to solve though, so we further tune this into a more manageable form.

Let the eigenvalue decomposition (EVD) of the estimated matrix $\hat{\mathbf{H}}$ be given by

$$\hat{\mathbf{H}} = \hat{\mathbf{U}} \operatorname{diag}(\hat{\boldsymbol{\lambda}})\hat{\mathbf{U}}^\top,$$

which leads to

$$\hat{\mathbf{H}}^2 = \hat{\mathbf{U}} \operatorname{diag}(\hat{\boldsymbol{\lambda}}^2)\hat{\mathbf{U}}^\top. \tag{3.13}$$

Notice this is naturally in an EVD form. Now the question transforms to estimating two variables $\hat{\mathbf{U}}$ and $\hat{\boldsymbol{\lambda}}$. Consider the EVD of $\mathbf{C}_{\mathbf{y}}$ that is given by

$$\mathbf{C}_{\mathbf{y}} = \mathbf{U}_{\mathbf{y}} \operatorname{diag}(\boldsymbol{\lambda}_{\mathbf{y}})\mathbf{U}_{\mathbf{y}}^\top$$

We can then approximate the covariance matching problem by setting $\hat{\mathbf{U}} = \mathbf{U}_{\mathbf{y}}$ and fitting $\hat{\boldsymbol{\lambda}}^2$ to $\boldsymbol{\lambda}_{\mathbf{y}}$. So the problem becomes

$$\begin{aligned} \boldsymbol{\lambda}^* = \arg\min_{\hat{\boldsymbol{\lambda}}} \quad & \|\hat{\boldsymbol{\lambda}}^2 - \boldsymbol{\lambda}_{\mathbf{y}}\|_2^2 \\ \text{subject to} \quad & \operatorname{diag}(\mathbf{U}_{\mathbf{y}}\operatorname{diag}(\hat{\boldsymbol{\lambda}}^{-1})\mathbf{U}_{\mathbf{y}}^\top) = \mathbf{1}. \end{aligned} \tag{3.14}$$

To solve the above question, we firstly switch the objective and constraint. Then the new optimization problem is formulated as

$$\begin{aligned} \boldsymbol{\lambda}^* = \arg\min_{\hat{\boldsymbol{\lambda}}} \quad & \left\|\operatorname{diag}(\mathbf{U}_{\mathbf{y}}\operatorname{diag}(\hat{\boldsymbol{\lambda}}^{-1})\mathbf{U}_{\mathbf{y}}^\top) - \mathbf{1}\right\|_2^2 \\ \text{subject to} \quad & \hat{\boldsymbol{\lambda}}^2 = \boldsymbol{\lambda}_{\mathbf{y}} \end{aligned} \tag{3.15}$$

You might think we have already obtained the solution, merely needing to take a square root of $\boldsymbol{\lambda}_{\mathbf{y}}$, but unfortunately, $\boldsymbol{\lambda}_{\mathbf{y}}$ can include negative values. Now that we can determine the absolute values of $\hat{\boldsymbol{\lambda}}$, the uncertainty remains only in their signs. To address this uncertainty, we introduce a variable $\hat{\mathbf{q}}$ to represent the signs of each element in $\hat{\boldsymbol{\lambda}}$. Let $\hat{\mathbf{q}} \in \{-1,1\}^{N\times 1}$, and $\hat{\boldsymbol{\lambda}} = \operatorname{diag}(\hat{\mathbf{q}})\boldsymbol{\lambda}_{\mathbf{y}}^{1/2}$. Rewriting the objective we then obtain

$$\begin{aligned}
&\|\operatorname{diag}(\mathbf{U}_{\mathbf{y}}\operatorname{diag}(\hat{\boldsymbol{\lambda}}^{-1})\mathbf{U}_{\mathbf{y}}^{\top}) - \mathbf{1}\|_2^2 \\
=\ &\|\operatorname{diag}(\mathbf{U}_{\mathbf{y}}\operatorname{diag}(\hat{\mathbf{q}}^{-1})\operatorname{diag}(\boldsymbol{\lambda}_{\mathbf{y}}^{-1/2})\mathbf{U}_{\mathbf{y}}^{\top}) - \mathbf{1}\|_2^2 \\
=\ &\|\operatorname{diag}(\mathbf{U}_{\mathbf{y}}\operatorname{diag}(\hat{\mathbf{q}})\operatorname{diag}(\boldsymbol{\lambda}_{\mathbf{y}}^{-1/2})\mathbf{U}_{\mathbf{y}}^{\top}) - \mathbf{1}\|_2^2 \\
=\ &\|(\mathbf{U}_{\mathbf{y}} \odot \mathbf{U}_{\mathbf{y}})\operatorname{diag}(\boldsymbol{\lambda}_{\mathbf{y}}^{-1/2})\hat{\mathbf{q}} - \mathbf{1}\|_2^2,
\end{aligned} \tag{3.16}$$

where we used that $\hat{\mathbf{q}}^{-1} = \hat{\mathbf{q}}$. Finally, defining

$$\mathbf{W} = (\mathbf{U}_{\mathbf{y}} \odot \mathbf{U}_{\mathbf{y}})\operatorname{diag}(\boldsymbol{\lambda}_{\mathbf{y}}^{-1/2})$$

the objective function becomes

$$\|\mathbf{W}\hat{\mathbf{q}} - \mathbf{1}\|_2^2$$

Hence, the final problem translates into

Optimization Problem 3.

$$\mathbf{q}^* = \arg\min_{\hat{\mathbf{q}}\in\{-1,1\}^{N\times 1}} \|\mathbf{W}\hat{\mathbf{q}} - \mathbf{1}\|_2^2$$

This problem can be solved by the SDR trick we disscussed in Section 2.5.

3.2.3 Identifiability Analysis

Proving directly whether our method is identifiable poses a significant challenge; however, we can establish that for a specific class of graphs, our estimated result $\mathbf{S}^* = \mathbf{I} - \mathbf{U}_{\mathbf{x}}\operatorname{diag}(\mathbf{q}^*)\operatorname{diag}(\boldsymbol{\lambda}_{\mathbf{x}}^{-1/2})\mathbf{U}_{\mathbf{x}}^{\top}$ from Optimization Problem 3 converges to the true $\mathbf{S}$. It is also important to note that SDR is a powerful approach to solve Optimization Problem 3, as discussed in [24].

Theorem 4. Let the EVD of the true GSO $\mathbf{S}$ be given by $\mathbf{S} = \mathbf{U}\operatorname{diag}(\boldsymbol{\lambda})\mathbf{U}^{\top}$. Further assume $\hat{\mathbf{p}}$ is a binary variable and consider the equation

$$(\mathbf{U} \odot \mathbf{U})|\mathbf{I} - \operatorname{diag}(\boldsymbol{\lambda})|\hat{\mathbf{p}} - \mathbf{1} = \mathbf{0}. \tag{3.17}$$

If this equation only has one binary solution $\mathbf{p}^*$, then the estimator $\mathbf{S}^*$, obtained from the solution of problem Optimization Problem 3, i.e., $\mathbf{S}^* = \mathbf{I} - \mathbf{U}_{\mathbf{x}}\operatorname{diag}(\mathbf{q}^*)\operatorname{diag}(\boldsymbol{\lambda}_{\mathbf{x}}^{-1/2})\mathbf{U}_{\mathbf{x}}^{\top}$, will converge to the true $\mathbf{S}$ when the number of observations T goes to infinity.

Our proof sketch starts with observing that at $T = \infty$ we have $\mathbf{C}_{\mathbf{x}} = \boldsymbol{\Sigma}_{\mathbf{x}} = (\mathbf{I} - \mathbf{S})^{-2}$. As a result, the EVD of $\mathbf{C}_{\mathbf{x}}$ then is $\mathbf{C}_{\mathbf{x}} = \mathbf{U}(\mathbf{I} - \operatorname{diag}(\boldsymbol{\lambda}))^{-2}\mathbf{U}^{\top}$, and thus $\mathbf{U}_{\mathbf{x}} = \mathbf{U}$ and $\boldsymbol{\lambda}_{\mathbf{x}}^{-1/2} = |\mathbf{1} - \operatorname{diag}(\boldsymbol{\lambda})|$. Hence, saying that (3.17) has a unique binary solution $\mathbf{p}^*$ is the same as saying that the optimization problem only has a unique solution $\mathbf{q}^*$ at $T = \infty$ and these solutions are then also the same.

Although this theorem may seem evident, it can be considered as a broadening of the theorem mentioned in [14], where $\operatorname{rank}(\mathbf{U} \odot \mathbf{U}) = N - 1$ is required. Under this condition, our theorem holds trivially.

However, our theorem has the potential to handle cases where $\mathrm{rank}(\mathbf{U} \odot \mathbf{U}) < N - 1$ and in our experiments, we will verify this in our last experiment.

3.3 Extension to General Distribution

In this section, we explore extending the distribution of the latent variable $\mathbf{e}$. More specifically, we assume that $\mathbf{e} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\Sigma}_{\mathbf{e}})$.

Estimating $\mathbf{S}$ then again boils down to estimating $\mathbf{H} = (\mathbf{I} - \mathbf{S})^{-1}$. At first sight, one could exploit the fact that $\boldsymbol{\Sigma}_{\mathbf{x}} = \mathbf{H}\boldsymbol{\Sigma}_{\mathbf{e}}\mathbf{H}^{\top}$ and match $\hat{\mathbf{H}}\boldsymbol{\Sigma}_{\mathbf{e}}\hat{\mathbf{H}}^{\top}$ with $\mathbf{C}_{\mathbf{x}}$. Solving this matching problem is challenging though. As an alternative, observe that $(\mathbf{H}\boldsymbol{\Sigma}_{\mathbf{e}})^2 = \mathbf{H}\boldsymbol{\Sigma}_{\mathbf{e}}\mathbf{H}\boldsymbol{\Sigma}_{\mathbf{e}} = \boldsymbol{\Sigma}_{\mathbf{x}}\boldsymbol{\Sigma}_{\mathbf{e}}$. This allows us to match $(\hat{\mathbf{H}}\boldsymbol{\Sigma}_{\mathbf{e}})^2$ with $\mathbf{C}_{\mathbf{x}}\boldsymbol{\Sigma}_{\mathbf{e}}$ which is similar to (3.12), where we matched $\hat{\mathbf{H}}^2$ with $\mathbf{C}_{\mathbf{x}}$. As a result, we follow again the same steps.

First, we introduce two new variables $\hat{\mathbf{U}}$ and $\hat{\boldsymbol{\lambda}}$ by considering the EVD of $\hat{\mathbf{H}}\boldsymbol{\Sigma}_{\mathbf{e}}$, i.e., $\hat{\mathbf{H}}\boldsymbol{\Sigma}_{\mathbf{e}} = \hat{\mathbf{U}}\,\mathrm{diag}(\hat{\boldsymbol{\lambda}})\hat{\mathbf{U}}^{-1}$. This obviously leads to $(\hat{\mathbf{H}}\boldsymbol{\Sigma}_{\mathbf{e}})^2 = \hat{\mathbf{U}}\,\mathrm{diag}(\hat{\boldsymbol{\lambda}})^2\hat{\mathbf{U}}^{-1}$. Computing the EVD of $\mathbf{C}_{\mathbf{x}}\boldsymbol{\Sigma}_{\mathbf{e}}$, we obtain[1] $\mathbf{C}_{\mathbf{x}}\boldsymbol{\Sigma}_{\mathbf{e}} = \mathbf{U}_{\mathbf{xe}}\,\mathrm{diag}(\boldsymbol{\lambda}_{\mathbf{xe}})\mathbf{U}_{\mathbf{xe}}^{-1}$. Setting now $\hat{\mathbf{U}} = \mathbf{U}_{\mathbf{xe}}$ and replacing the matching problem by the constraint $\hat{\boldsymbol{\lambda}}^2 = \boldsymbol{\lambda}_{\mathbf{xe}}$ introduces once again a sign ambiguity. More specifically, we can change the variable $\hat{\boldsymbol{\lambda}}$ by the binary variable $\hat{\mathbf{q}} \in \{-1, 1\}^{N\times 1}$ using $\hat{\boldsymbol{\lambda}} = \mathrm{diag}(\hat{\mathbf{q}})\boldsymbol{\lambda}_{\mathbf{xe}}^{1/2}$. Overall, this allows us to write $\hat{\mathbf{H}}$ as a function of $\hat{\mathbf{q}}$ through

$$\hat{\mathbf{H}} = \mathbf{U}_{\mathbf{xe}}\,\mathrm{diag}(\hat{\mathbf{q}})\mathrm{diag}(\boldsymbol{\lambda}_{\mathbf{xe}}^{1/2})\mathbf{U}_{\mathbf{xe}}^{-1}\boldsymbol{\Sigma}_{\mathbf{e}}^{-1}. \tag{3.18}$$

The inverse of $\hat{\mathbf{H}}$ is then given by

$$\hat{\mathbf{H}}^{-1} = \boldsymbol{\Sigma}_{\mathbf{e}}\mathbf{U}_{\mathbf{xe}}\mathrm{diag}(\boldsymbol{\lambda}_{\mathbf{xe}}^{-1/2})\,\mathrm{diag}(\hat{\mathbf{q}})\mathbf{U}_{\mathbf{xe}}^{-1} \tag{3.19}$$

and the diagonal of $\hat{\mathbf{H}}^{-1}$ is

$$\mathrm{diag}(\hat{\mathbf{H}}^{-1}) = [(\boldsymbol{\Sigma}_{\mathbf{e}}\mathbf{U}_{\mathbf{xe}}) \odot \mathbf{U}_{\mathbf{xe}}^{-\top}]\mathrm{diag}(\boldsymbol{\lambda}_{\mathbf{xe}}^{-1/2})\hat{\mathbf{q}}. \tag{3.20}$$

Finally, defining $\mathbf{W} = [(\boldsymbol{\Sigma}_{\mathbf{e}}\mathbf{U}_{\mathbf{xe}}) \odot \mathbf{U}_{\mathbf{xe}}^{-\top}]\mathrm{diag}(\boldsymbol{\lambda}_{\mathbf{xe}}^{-1/2})$, the optimization problem simplifies to:

Optimization Problem 4.

$$\min_{\hat{\mathbf{q}} \in \{1,1\}^{N\times 1}} \|\mathbf{W}\hat{\mathbf{q}} - \mathbf{1}\|_2^2, \tag{3.21}$$

which is again a BQP that can be solved using semi-definite relaxation.

Comparing Optimization Problem 3 and Optimization Problem 4, their formulations are almost identical. Furthermore, if we assume $\boldsymbol{\Sigma}_{\mathbf{e}} = \mathbf{I}$, then $\mathbf{U}_{\mathbf{xe}}$ in Optimization Problem 4 reduces to $\mathbf{U}_{\mathbf{x}}$, and $\mathbf{U}_{\mathbf{xe}}^{-\top}$ also reduces to $\mathbf{U}_{\mathbf{x}}$. Moreover, $\boldsymbol{\lambda}_{\mathbf{xe}}^{-1/2}$ reduces to $\boldsymbol{\lambda}_{\mathbf{x}}^{-1/2}$. This extension of the SEM problem is quite elegant, as it scarcely alters the structure of the problem, and the complexity of solving the optimization problem remains the same.

[1] Note that we use the notation $\mathbf{U}$ primarily to align with the previous notation and it does not imply that $\mathbf{U}$ is unitary.

3.4 Experiments

3.4.1 Graph Configuration

In order to demonstrate the superiority of our method, we selected two challenging graph configurations for our experiments.

Fully Connected Graph: First, we generated a fully connected graph, which corresponds to not making any sparsity assumptions. Specifically, we constructed a graph with 20 nodes where every node is connected to every other node. The weights of the edges are assigned values in the range $[-2, -0.1] \bigcup [0.1, 2]$.

Sparse Graph: The second configuration involved generating a sparse graph. For this setup, we chose to constrain the rank of $\mathbf{S}$ to $N - 3$. Specifically, the graph consisted of 20 nodes with 20 edges, and the edge weights were also set within the range $[-2, -0.1] \bigcup [0.1, 2]$.

The experiments will demonstrate that our method can effectively address three key issues that other methods struggle with:

- **Non-sparse issues:** Virtually all methods assume some level of sparsity [14, 2].
- **Negative edge values:** This issue, which is unresolved by methods such as those in [2], poses significant challenges.
- **Rank constraints ($\leq N-2$):** Problems arise when $\mathbf{\Sigma_y}$ has repeated eigenvalues, at which point methods relying on commutative constraints [14, 2] fail.

To configure the matrix $\mathbf{\Sigma_e}$ in case the latent variables are not white, we first generated a random square root matrix, denoted as $\mathbf{\Sigma}_{\text{sqrt}}$, which is an $N \times N$ matrix with each element uniformly distributed in the range [-1, 1]. The matrix $\mathbf{\Sigma_e}$ was then formed by $\mathbf{\Sigma}_{\text{sqrt}}\mathbf{\Sigma}_{\text{sqrt}}^{\top}$, ensuring that $\mathbf{\Sigma_e}$ is symmetric and positive definite, which is crucial for the stability of the graph-based processes studied.

Beyond addressing these issues, our method offers numerous additional advantages, such as requiring no parameter tuning, eliminating any scale ambiguity and handling any latent Gaussian variables.

For each experimental setup, we provide two types of results. First, results for a sample dataset that demonstrate what the estimation should look like for a specific graph under ideal conditions ($T \to \infty$) and when $T = 10000$. This highlights the behavior of our method for individual graph instances.

Secondly, we report results averaged over a larger set of graphs. Specifically, we randomly generate 100 graphs meeting the described criteria and we measure the distribution of errors as the number of samples changes. We utilize box plots to visually represent these results, providing a clear overview of the error distribution across different scenarios.

In our final experiment, we compare our method against PGL [2], SpecTemp [14] and SigMatch [6].

For assessing the error, we employ the normalized square error (NSE) to quantify discrepancies between the estimated and the true graph structures. Specifically, given the estimated graph $\mathbf{S}^*$ and the true graph $\mathbf{S}$, the error is measured using the following formula:

$$\text{NSE}\left(\mathbf{S}^*, \mathbf{S}\right) = \frac{\|\mathbf{S}^* - \mathbf{S}\|_F^2}{\|\mathbf{S}\|_F^2} \tag{3.22}$$

3.4.2 One-Sample Results for Sparse and Fully Connected Graphs

We begin by examining individual sample results for both sparse and fully connected graph configurations. Fig. 3.1 illustrates the true graph configuration for the sparse graph experiment, while Fig. 3.2 shows the true graph configuration for the fully connected graph experiment. These figures provide a visual representation of the estimated graphs under both ideal and practical conditions, allowing us to assess the effectiveness of our method in different scenarios.

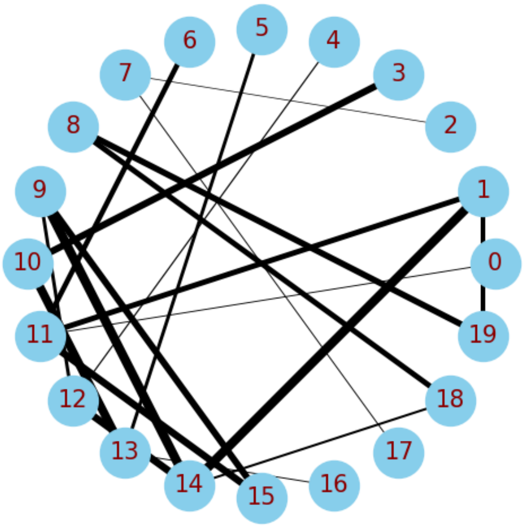


Figure 3.1: True graph configuration for the sparse graph experiment.

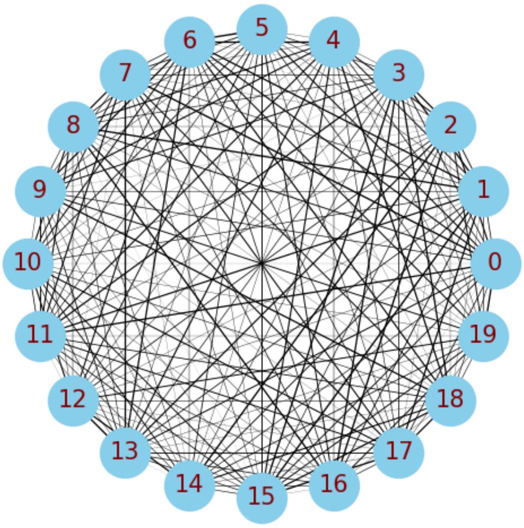


Figure 3.2: True graph configuration for the fully connected graph experiment.

Results under ideal conditions ($T \to \infty$) for both configurations are demonstrated in Fig. 3.3 for the sparse graph and Fig. 3.4 for the fully connected graph. In these scenarios, the normalized square error (NSE) is smaller than 10^{-10}, indicating highly accurate graph estimation under ideal conditions.

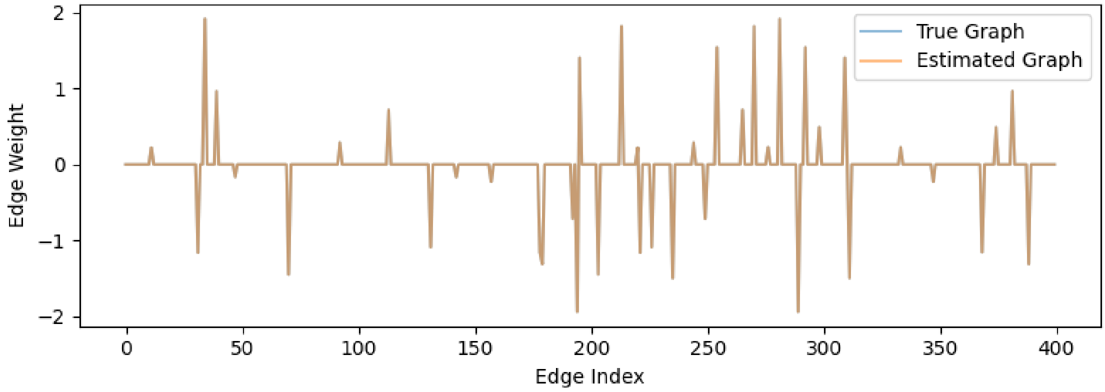


Figure 3.3: Estimated sparse graph under ideal conditions.

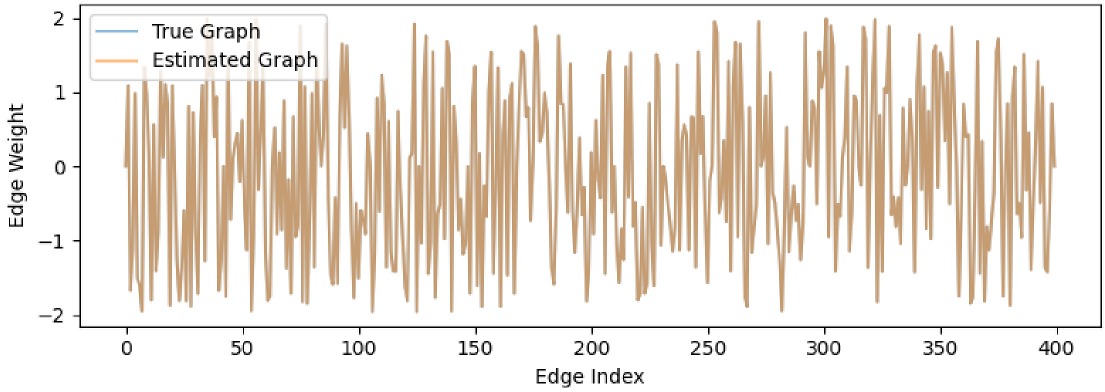


Figure 3.4: Estimated fully connected graph under ideal conditions.

For the finite observation case with $T = 10000$ samples, the results are depicted in Fig. 3.5 for the sparse graph and Fig. 3.6 for the fully connected graph. Here, the NSE for the sparse graph is 2.591×10^{-2} and for the fully connected graph, it is 1.209×10^{-1}.

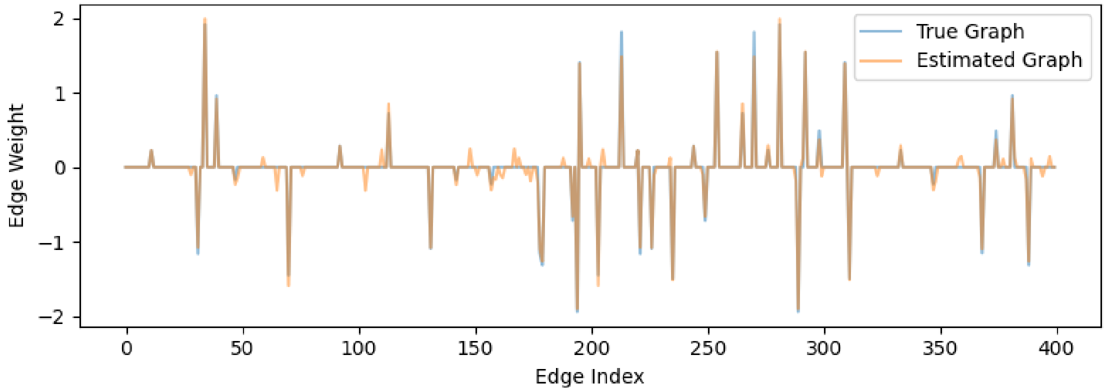


Figure 3.5: Estimated sparse graph under practical conditions. NSE: 2.591×10^{-2}.

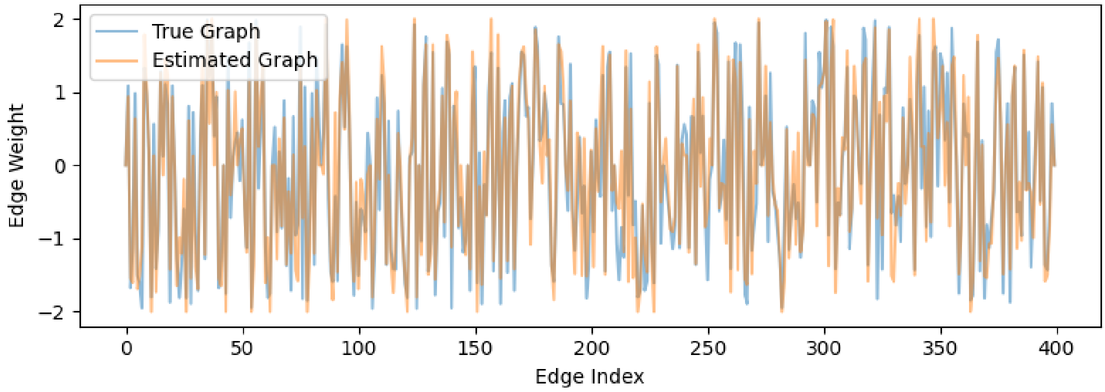


Figure 3.6: Estimated fully connected graph under practical conditions. NSE: 1.209×10^{-1}.

For sparse graphs, our estimations continue to be quite accurate. For fully connected graphs, the results are acceptable, which also suggests that fully connected graphs indeed represent a more challenging problem. In fact, addressing the challenges posed by dense graphs was the initial motivation for developing our method, given that other methods yield poor results for such graph configurations.

3.4.3 Average Results Across 100 Graphs

The relationship between the number of samples and the distribution of the estimation error is depicted as follows

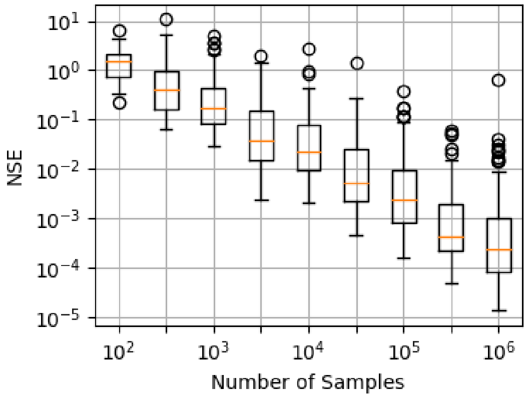


Figure 3.7: Boxplot showing the distribution of NSE as a function of sample size for sparse graphs.

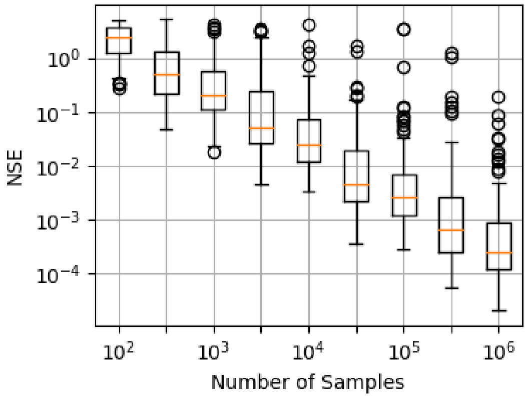


Figure 3.8: Boxplot showing the distribution of NSE for fully connected graphs as a function of sample size.

As $T \to \infty$, both configurations exhibit a decrease in NSE, demonstrating the effectiveness of our method under varying graph densities and structures.

3.4.4 Comparision with Other Methods

Below we present a comparison of our results with those from [2]. Fig. 3.9 shows the PGL's performance across different graph types: small world (SW), stochastic block model (SBM), and Barabási-Albert (BA) graphs. It is important to note that in PGL's context, SSEM pertains to the undirected graph SEM, which is also the focus of our study. At the same time, we adopt the simplest assumption that $\mathbf{\Sigma_e} = \mathbf{I}$.

Unlike PGL, we did not restrict our optimization problem to non-negative weights, which is a prerequisite for PGL's methodology. Fig. 3.9 and Fig. 3.10 respectively showcase the results from PGL and our CovMatch approach. In Fig. 3.9, SSEM refers to symmetric SEM, where the symmetry of the graph aligns with the assumptions made in our analysis. This mirrors the symmetric structure of the graph, as considered in our scenario.

Despite the broader constraints in our method, our results for the Small World (SW) and Stochastic Block Model (SBM) models surpass those reported by PGL. As for the Barabási-Albert (BA) model, we conjecture that the exceptionally good performance noted in [2] may be due to randomness. Our results show no significant distinctions across the three graph models, while the remarkable results for the BA model in [2] are counterintuitive.

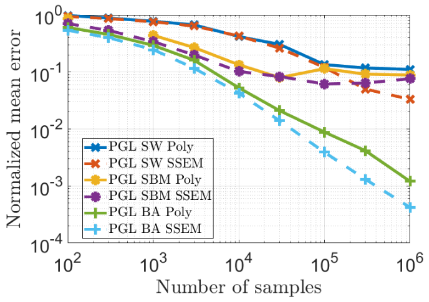


Figure 3.9: Performance results from PGL [2] on SW, SBM, and BA models, demonstrating the effectiveness of non-negative weight constraints.

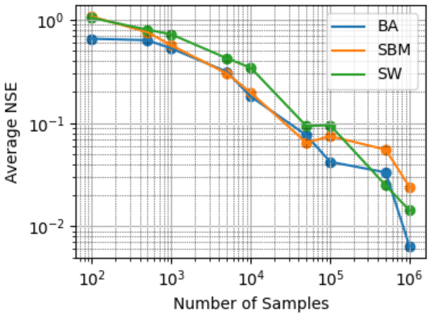


Figure 3.10: Results from CovMatch indicating average Normalized Square Error (NSE) across SW, SBM, and BA models, showcasing performance without non-negative weight constraints.

We also compare our approach (refered to as CovMatch) with SpecTemp [14] and with a trivial signal matching approach (SigMatch) [6] based on minimizing $\|\mathbf{X}-\hat{\mathbf{S}}\mathbf{X}\|_F^2$. Due to the sign constraints of SpecTemp, all graphs are assigned positive weights ranging from 0.1 to 2. At the same time, we adopt the simplest assumption that $\mathbf{\Sigma_e} = \mathbf{I}$. In the first experiment (labelled as simple), we deliberately generate scenarios where $\text{rank}(\mathbf{U} \odot \mathbf{U}) = N - 1$, using graphs with 20 nodes and 20 edges. In the second experiment (labelled as hard), we only generate graphs with $\text{rank}(\mathbf{U} \odot \mathbf{U}) < N - 1$ to check the robustness of our method under less controlled conditions. Over the 100 graph realizations, we calculate the average NSE. Note that a singular value less than 5×10^{-4} is considered as a rank loss here.

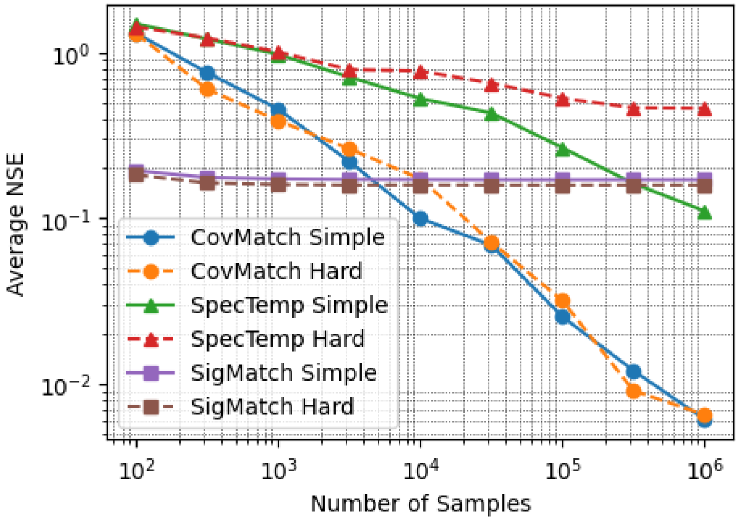


Figure 3.11: Average NSE for different samples.

As shown in Fig. 3.11, it is evident that SpecTemp often fails due to a loss of rank.

Conversely, our method, CovMatch, continues to perform well. This highlights the robustness and reliability of CovMatch. Further, the SigMatch approach never converges to the correct result, but it performs better than others with less observations. This is because both SpecTemp and CovMatch highly rely on an accurate sample covariance, which requires many samples.

Extension to Directed Graphs 4

In this chapter, we address a more challenging problem: can we identify $\mathbf{S}$ corresponding to directed graphs in a SEM using covariance matching? Our discussion is structured as follows: we first analyze why this problem is particularly difficult, primarily due to the issue of having more variables than equations. Consequently, we do not aim to solve the general problem outright; instead, we propose that by imposing a sparsity constraint, a fundamental optimization problem can be derived. We then discuss how this optimization problem can be relaxed to a convex optimization problem. Additionally, we will introduce the use of the Schur complement as a method to make the problem solvable by cvx [30]. This chapter also serves as a preparation for the next chapter. We will not present experimental results here, as a more unified method will be introduced and experimentally validated in the following chapter.

4.1 Challenges for a SEM based on Directed Graphs

Before discussing how to generalize our approach to directed graphs, it is crucial to understand why this task is inherently more difficult.

Assume we again consider the standard SEM where

$$\mathbf{y} = (\mathbf{I} - \mathbf{S})^{-1}\mathbf{e}, \quad \mathbf{e} \sim N(0, \mathbf{I}), \tag{4.1}$$

yet now $\mathbf{S}$ is a non-symmetric but hollow matrix.

Again, let $\mathbf{H} = (\mathbf{I} - \mathbf{S})^{-1}$, which is also a non-symmetric matrix. Assuming the SVD of $\mathbf{H}$ is $\mathbf{U}\boldsymbol{\Lambda}\mathbf{V}^\top$, we obtain

$$\boldsymbol{\Sigma}_\mathbf{y} = \mathbb{E}[\mathbf{y}\mathbf{y}^T] = \mathbf{H}\mathbf{H}^\top = \mathbf{U}\boldsymbol{\Lambda}^2\mathbf{U}^\top. \tag{4.2}$$

As previously discussed, for Gaussian distributions, knowing the variance and mean equates to knowing all information about $\mathbf{y}$. However, from the result above, it is apparent that we invariably lose all information about $\mathbf{V}$. It is also worthy to note that there is no sign ambiguity on $\boldsymbol{\Lambda}$ here, because we are utilizing the SVD, which ensures $\boldsymbol{\Lambda} \geq \mathbf{0}$. Therefore, all uncertainties are concentrated in the unitary matrix $\mathbf{V}$.

Now, let us consider whether we have sufficient information to solve for $\mathbf{V}$. In the general case, for any $\mathbf{S}$, it is impossible to determine the corresponding $\mathbf{V}$.

Considering all elements of the unitary matrix, we have N^2 variables. The constraints of unitarity equate to $\frac{N(N-1)}{2}$ orthogonal conditions plus N conditions for vector norms to be 1. The only remaining constraints are the diagonal constraints, which are N equations. Thus, we have:

$$\frac{N(N-1)}{2} + N + N = \frac{N(N+3)}{2}$$

equations. When $N > 3$, $\frac{N(N+3)}{2} < N^2$, implying that the number of unknowns exceeds the number of equations, rendering the graph unidentifiable.

While solving for the general directed graph in the context of a SEM is impractical, it is important to recognize that graphs in real-life applications are typically sparse. This means that most edge weights are zero or close to zero. Hence, including a sparsity constraint, we might be able to boost the number of equations and correctly estimate the graph. In fact, our experiments have demonstrated that correct results can be obtained when the graph is sparse.

4.2 Covariance Matching with Sparsity

Building on the challenges described earlier, we incorporate the sparsity information of the graph to formulate our new optimization problem. We assume a constraint that allows the sparsity level of $\hat{\mathbf{S}}$, measured by the ℓ_0-norm, to be less than or equal to $s_{\max}$. As before, we want to match $\hat{\mathbf{H}}\hat{\mathbf{H}}^\top$ and $\mathbf{C}_\mathbf{y} = \frac{\mathbf{Y}\mathbf{Y}^\top}{T}$. This leads to the following problem

$$\begin{aligned} \mathbf{H}^* = \arg\min_{\hat{\mathbf{H}}} \quad & \|\hat{\mathbf{H}}\hat{\mathbf{H}}^\top - \mathbf{C}_\mathbf{y}\|_F \\ \text{subject to} \quad & \operatorname{diag}(\hat{\mathbf{H}}^{-1}) = \mathbf{1}, \\ & \|\mathbf{I} - \hat{\mathbf{H}}^{-1}\|_0 \leq s_{max} \end{aligned} \tag{4.3}$$

To estimate $\mathbf{H}$, assume the EVD of $\mathbf{C}_\mathbf{y}$ is given by $\mathbf{C}_\mathbf{y} = \mathbf{U}_\mathbf{y} \operatorname{diag}(\boldsymbol{\lambda}_\mathbf{y})\mathbf{U}_\mathbf{y}^\top$. The SVD of $\hat{\mathbf{H}}$ is denoted as $\hat{\mathbf{U}} \operatorname{diag}(\hat{\boldsymbol{\lambda}})\hat{\mathbf{V}}^\top$, thereby introducing three new variables, $\hat{\mathbf{U}}$, $\hat{\boldsymbol{\lambda}}$, and $\hat{\mathbf{V}}$. Further observe that $\hat{\mathbf{H}}\hat{\mathbf{H}}^\top = \hat{\mathbf{U}} \operatorname{diag}(\hat{\boldsymbol{\lambda}}^2)\hat{\mathbf{U}}^\top$, which itself is in the form of an EVD. To approximate the solution of the earlier problem, we now set $\hat{\mathbf{U}} = \mathbf{U}_\mathbf{y}$ and $\hat{\boldsymbol{\lambda}} = \boldsymbol{\lambda}_\mathbf{y}^{1/2}$, which means $\hat{\mathbf{H}} = \mathbf{U}_\mathbf{y} \operatorname{diag}(\boldsymbol{\lambda}_\mathbf{y}^{1/2})\hat{\mathbf{V}}^\top$ can be represented by the single unitary variable $\hat{\mathbf{V}}$. Rewriting the diagonal constraint as objective, we then obtain the problem

$$\begin{aligned} \arg\min_{\hat{\mathbf{V}}} \quad & \left\| \operatorname{diag}(\hat{\mathbf{V}} \operatorname{diag}(\boldsymbol{\lambda}_\mathbf{y}^{-1/2})\mathbf{U}_\mathbf{y}^\top - \mathbf{I}) \right\|_F \\ \text{subject to:} \quad & \hat{\mathbf{V}}\hat{\mathbf{V}}^\top = \mathbf{I}, \\ & \|\mathbf{I} - \hat{\mathbf{V}} \operatorname{diag}(\boldsymbol{\lambda}_\mathbf{y}^{-1/2})\mathbf{U}_\mathbf{y}^\top\|_0 \leq s_{\max} \end{aligned} \tag{4.4}$$

This optimization problem still presents two challenges. The first is the difficulty in optimizing the L_0 norm, which is inherently non-convex and hard to handle directly. As an approximation, we can transform the L_0 norm into an L_1 norm and incorporate it into the objective function. This leads to

$$
\begin{aligned}
\arg\min_{\hat{\mathbf{V}}} \quad & \left\|\operatorname{diag}(\hat{\mathbf{V}} \operatorname{diag}(\boldsymbol{\lambda}_{\mathbf{y}}^{-1/2})\mathbf{U}_{\mathbf{y}}^{\top} - \mathbf{I})\right\|_F + \alpha \left\|\hat{\mathbf{V}} \operatorname{diag}(\boldsymbol{\lambda}_{\mathbf{y}}^{-1/2})\mathbf{U}_{\mathbf{y}}^{\top} - \mathbf{I}\right\|_1 \\
\text{subject to:} \quad & \hat{\mathbf{V}}\hat{\mathbf{V}}^{\top} = \mathbf{I}
\end{aligned}
\tag{4.5}
$$

The second issue concerns the non-convex space formed by unitary matrices. In this chapter, we propose a relaxation method to address this challenge. In the following chapter, we will introduce a more robust solution that combines this relaxation approach with a projection.

4.2.1 Convex Relaxation for Unitary Variable

In our optimization framework, the sole non-convex constraint is $\hat{\mathbf{V}}\hat{\mathbf{V}}^{\top} = \mathbf{I}$. To simplify this, we propose relaxing it to $\hat{\mathbf{V}}\hat{\mathbf{V}}^{\top} \preceq \mathbf{I}$, thus creating a convex feasible space. Although it is not immediately obvious, we will demonstrate next why this relaxation leads to a convex space. Note that this modification is akin to the scalar scenario where the region of x constrained by $x^2 = 1$ is non-convex, but the region of x constrained by $x^2 \leq 1$ is convex.

Using this relaxation technique, we can finally rewrite our problem as
Optimization Problem 5.

$$
\begin{aligned}
\arg\min_{\hat{\mathbf{V}}} \quad & \left\|\operatorname{diag}(\hat{\mathbf{V}} \operatorname{diag}(\boldsymbol{\lambda}_{\mathbf{y}}^{-1/2})\mathbf{U}_{\mathbf{y}}^{\top} - \mathbf{I})\right\|_F + \alpha \left\|\hat{\mathbf{V}} \operatorname{diag}(\boldsymbol{\lambda}_{\mathbf{y}}^{-1/2})\mathbf{U}_{\mathbf{y}}^{\top} - \mathbf{I}\right\|_1 \\
\text{subject to:} \quad & \hat{\mathbf{V}}\hat{\mathbf{V}}^{\top} \preceq \mathbf{I}
\end{aligned}
\tag{4.6}
$$

4.2.1.1 Proof of Convexity for Relaxed Constraint

To prove the convexity of the constraint $\hat{\mathbf{V}}\hat{\mathbf{V}}^{\top} \preceq \mathbf{I}$, we introduce the Schur complement and the lemma [31] :
Lemma 3 (Schur Complement Condition)**.** For any symmetric matrix $\mathbf{M}$ of the form

$$
\mathbf{M} = \begin{pmatrix} \mathbf{A} & \mathbf{B} \\ \mathbf{B}^{\top} & \mathbf{C} \end{pmatrix}, \tag{4.7}
$$

if $\mathbf{C} \succ \mathbf{0}$, then $\mathbf{M} \succeq \mathbf{0}$ if and only if $\mathbf{A} - \mathbf{B}\mathbf{C}^{-1}\mathbf{B}^{\top} \succeq \mathbf{0}$.

Using the Schur complement lemma, we set $\mathbf{A} = \mathbf{C} = \mathbf{I}$ and $\mathbf{B} = \hat{\mathbf{V}}$ in the lemma's framework, yielding the matrix

$$
\bar{\mathbf{V}} = \begin{pmatrix} \mathbf{I} & \hat{\mathbf{V}} \\ \hat{\mathbf{V}}^{\top} & \mathbf{I} \end{pmatrix}.
$$

According to the lemma, since $\mathbf{I} \succ 0$, $\bar{\mathbf{V}} \succeq \mathbf{0}$ if and only if $\mathbf{I} - \hat{\mathbf{V}}\hat{\mathbf{V}}^{\top} \succeq 0$ (or $\hat{\mathbf{V}}\hat{\mathbf{V}}^{\top} \preceq \mathbf{I}$). As a result, $\hat{\mathbf{V}}\hat{\mathbf{V}}^{\top} \preceq \mathbf{I}$ characterizes a convex set, since the equivalent condition $\bar{\mathbf{V}} \succeq \mathbf{0}$ is a linear matrix inequality which always represents a convex set.

Introducing this property is beneficial for another reason. Although a convex region has been identified for the constraint $\hat{\mathbf{V}}\hat{\mathbf{V}}^\top \preceq \mathbf{I}$, a direct implementation in code is not feasible as such a constraint does not satisfy the disciplined convex programming (DCP) rules [32]. However, the configuration

$$\bar{\mathbf{V}} = \begin{pmatrix} \mathbf{I} & \hat{\mathbf{V}} \\ \hat{\mathbf{V}}^\top & \mathbf{I} \end{pmatrix} \succeq 0$$

adheres to DCP rules and can be solved using convex optimization tools like CVX.

Unified SEM Approach 5

Just as we extended the SEM for undirected graphs from white to non-white latent variable scenarios, we have long contemplated a similar expansion for the SEM using directed graphs. The excitement surrounding this prospect stems from its potential to complete our theoretical framework of the SEM. If successful, we would have a comprehensive model covering all permutations: SEM for white latent variables and undirected graphs, for non-white latent variables and undirected graphs, for white latent variables and directed graphs, and the final piece of the puzzle, for non-white latent variables and directed graphs.

In this chapter, we will introduce a new methodology robust enough to handle nearly all scenarios within a unified framework, regardless of whether the graph is directed or contains cycles. The only necessary condition is that $\mathbf{S}$ must be sparse. It is worthy to note this does not imply that previous discussions are entirely irrelevant. Since for undirected graphs, the methods previously discussed actually do not require the graph to be sparse.

5.1 Covariance Matching Optimization Problem

In this chapter, assume that $\mathbf{y}$ and $\mathbf{e}$ satisfy the regular SEM

$$\mathbf{y} = (\mathbf{I} - \mathbf{S})^{-1}\mathbf{e},$$

where $\mathbf{e} \sim \mathcal{N}(0, \mathbf{\Sigma_e})$ with $\mathbf{\Sigma_e}$ known but $\mathbf{e}$ unknown. The underlying graph here is known to be sparse, so $\mathbf{S}$ is a sparse matrix. Our goal is to derive $\mathbf{S}$ from $\mathbf{Y}$. Again, we denote the sample covariance by $\mathbf{C_y} = \frac{\mathbf{YY}^\top}{T}$ and let $\mathbf{H} = (\mathbf{I} - \mathbf{S})^{-1}$.

The covariance matrix of $\mathbf{y}$, $\mathbf{\Sigma_y}$, can be expressed as:

$$\mathbf{\Sigma_y} = \mathbf{E}\{\mathbf{yy}^\top\} = \mathbf{H\Sigma_e H}^\top. \tag{5.1}$$

Now, we aim to match the covariance matrix $\mathbf{C_y}$ with $\hat{\mathbf{H}}\mathbf{\Sigma_e}\hat{\mathbf{H}}^\top$. Similar to (4.3), we can formulate the problem

$$\begin{aligned} \mathbf{H}^* = \arg\min_{\hat{\mathbf{H}}} \quad & \|\hat{\mathbf{H}}\mathbf{\Sigma_e}\hat{\mathbf{H}}^\top - \mathbf{C_y}\|_F \\ \text{subject to} \quad & \operatorname{diag}(\hat{\mathbf{H}}^{-1}) = \mathbf{1}, \\ & \|\mathbf{I} - \hat{\mathbf{H}}^{-1}\|_0 \leq s_{max} \end{aligned} \tag{5.2}$$

To minimize the objective, we apply the Cholesky decomposition[1]. Given that both $\mathbf{C_y}$ and $\mathbf{\Sigma_e}$ are generally positive definite[2], they can be uniquely decomposed as follows

$$\begin{aligned} \mathbf{C_y} &= \mathbf{L_y}\mathbf{L_y}^\top, \\ \mathbf{\Sigma_e} &= \mathbf{L_e}\mathbf{L_e}^\top. \end{aligned} \tag{5.3}$$

Rewriting the objective, we obtain

$$\|\hat{\mathbf{H}}\mathbf{\Sigma_e}\hat{\mathbf{H}}^\top - \mathbf{C_y}\|_F = \|(\hat{\mathbf{H}}\mathbf{L_e})(\hat{\mathbf{H}}\mathbf{L_e})^\top - \mathbf{L_y}\mathbf{L_y}^\top\|_F.$$

Forcing this objective to zero, we can state that $\mathbf{L_y}$ and $\hat{\mathbf{H}}\mathbf{L_e}$ can differ only by a unitary matrix $\hat{\mathbf{V}}$, that is

$$\hat{\mathbf{H}}\mathbf{L_e} = \mathbf{L_y}\hat{\mathbf{V}}. \tag{5.4}$$

This allows us to replace the GSO variable $\hat{\mathbf{S}}$ by the unitary variable $\hat{\mathbf{V}}$ as

$$\hat{\mathbf{S}} = \mathbf{I} - \hat{\mathbf{H}}^{-1} = \mathbf{I} - \mathbf{L_e}\hat{\mathbf{V}}^\top\mathbf{L_y}^{-1}. \tag{5.5}$$

Observing that $\hat{\mathbf{S}}$ is linear with respect to $\hat{\mathbf{V}}$, which is similar to the conditions in (4.5), allows for the following problem formulation:

$$\begin{aligned} \mathbf{V}^* = \arg\min_{\hat{\mathbf{V}}} \quad & \left\|\operatorname{diag}(\mathbf{L_e}\hat{\mathbf{V}}^\top\mathbf{L_y}^{-1} - \mathbf{I})\right\|_F + \alpha\left\|\mathbf{L_e}\hat{\mathbf{V}}^\top\mathbf{L_y}^{-1} - \mathbf{I}\right\|_1 \\ \text{subject to:} \quad & \hat{\mathbf{V}}\hat{\mathbf{V}}^\top = \mathbf{I} \end{aligned} \tag{5.6}$$

5.2 Iterative Method

The primary reason for proposing an iterative method, is that $\hat{\mathbf{V}}^*$ often fails to satisfy the unitarity condition. To tackle this, our strategy involves re-mapping $\hat{\mathbf{V}}$ to a unitary matrix after each iteration. In subsequent iterations, the objective function is augmented with a term $\beta\|\hat{\mathbf{V}} - \hat{\mathbf{V}}_u\|_F$ to ensure that $\hat{\mathbf{V}}$ remains close to a unitary matrix, where $\hat{\mathbf{V}}_u$ is the unitary version of $\hat{\mathbf{V}}$ from a previous iteration.

5.2.1 Projection onto Unitary Matrix Space

To project a matrix $\hat{\mathbf{V}}$ onto the space of unitary matrices. Denote the projection result by $\hat{\mathbf{V}}_u$. This process is equivalent to solving an optimization problem formulated as:

$$\min_{\hat{\mathbf{V}}_u\hat{\mathbf{V}}_u^\top = \mathbf{I}} \|\hat{\mathbf{V}} - \hat{\mathbf{V}}_u\|_F. \tag{5.7}$$

This problem is equivalent to minimizing the Frobenius norm difference between $\hat{\mathbf{V}}\hat{\mathbf{V}}_u^\top$ and $\mathbf{I}$, which is expressed as:

$$\min_{\hat{\mathbf{V}}_u\hat{\mathbf{V}}_u^\top = \mathbf{I}} \|\hat{\mathbf{V}}\hat{\mathbf{V}}_u^\top - \mathbf{I}\|_F, \tag{5.8}$$

[1] We can also use the SVD, but the Cholesky decomposition is more concise and avoids excessive notation.

[2] The Cholesky decomposition is unique for positive definite matrices and exists for semidefinite matrices.

characterizing it as an orthogonal Procrustes problem [33].

The closed-form solution to this problem can be obtained by performing an SVD of $\hat{\mathbf{V}}$, given by $\hat{\mathbf{V}} = \mathbf{U}_v \mathbf{\Sigma}_v \mathbf{R}_v^\top$. And the unitary matrix $\mathbf{V}_u^*$ is then determined as

$$\mathbf{V}_u^* = \mathbf{U}_v \mathbf{R}_v^\top. \tag{5.9}$$

5.2.2 Algorithm Description

The following algorithm provides a detailed step-by-step procedure to implement the iterative method:

```
Algorithm 1 Iterative Method to Ensure Unitarity of $\mathbf{V}$
Require: Initial guess for $\mathbf{V}$, $\mathbf{V}^{(0)}$, tolerance $\epsilon$, maximum iterations $N$, weights $\alpha$, $\beta$
Ensure: $\mathbf{V}$ approximates a unitary matrix
  for $k = 1$ to $N$ do
    Step 1: Project $\mathbf{V}^{(k)}$ onto the unitary space to get $\mathbf{V}_u^{(k)}$
    Step 2: Solve the optimization problem for $\mathbf{V}^{(k+1)}$:
      $\mathbf{V}^{(k+1)} = \arg\min_{\hat{\mathbf{V}}} \left\| \mathrm{diag}(\mathbf{L}_{\mathbf{e}} \hat{\mathbf{V}}^\top \mathbf{L}_{\mathbf{y}}^{-1} - \mathbf{I}) \right\|_F + \alpha \left\| \mathbf{L}_{\mathbf{e}} \hat{\mathbf{V}}^\top \mathbf{L}_{\mathbf{y}}^{-1} - \mathbf{I} \right\|_1 + \beta \|\hat{\mathbf{V}} - \mathbf{V}_u^{(k)}\|_F$
    Step 3: Check for convergence:
    if $\|\mathbf{V}^{(k+1)} - \mathbf{V}_u^{(k)}\|_F < \epsilon$ then
      break
    end if
  end for
  return $\mathbf{V}^{k+1}$
```

This algorithm iteratively adjusts $\mathbf{V}$, refining it towards unitarity, thereby enhancing the alignment of the solution with the required mathematical properties.

5.3 Experiment

In this chapter, we firstly validate our method on graphs with cycles to demonstrate its effectiveness. In contrast to Section 3.4, we have not conducted experiments on a large number of graphs. This limitation is primarily due to the high computational complexity of our algorithm, which makes extensive testing time-consuming and resource-intensive. Despite these limitations, we provide a comparative analysis with the results from [9], using graph structures generated from their code. The outcomes show that our method is more accurate than the one proposed in [9].

5.3.1 Experiment on Directed Graphs with Cycles

A random directed graph with 15 nodes and 15 edges is generated repeatedly until a graph containing at least one cycle is obtained. If a generated graph is acyclic, it is discarded and the process is retried. The obtained graph from Fig. 5.1 has two cycles:

$9 \to 10 \to 7 \to 2 \to 9$, $1 \to 13 \to 4 \to 1$. Additionally, we generate $\boldsymbol{\Sigma}_\mathbf{x}$ following the method outlined in Subsection 3.4.1 for generating $\boldsymbol{\Sigma}_\mathbf{e}$.

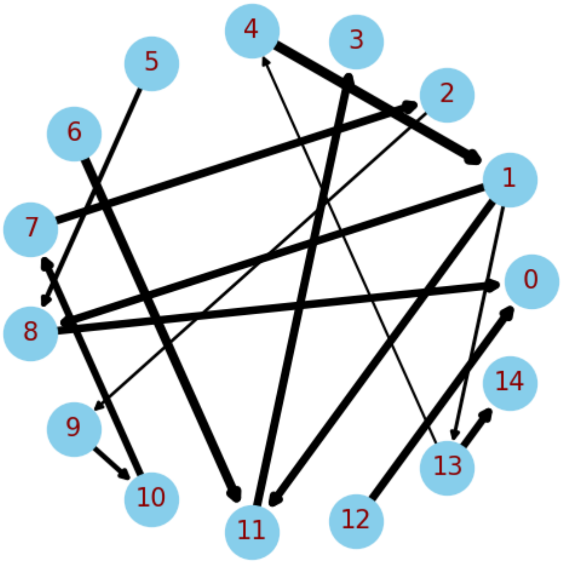


Figure 5.1: True graph configuration for a directed graph with cycles.

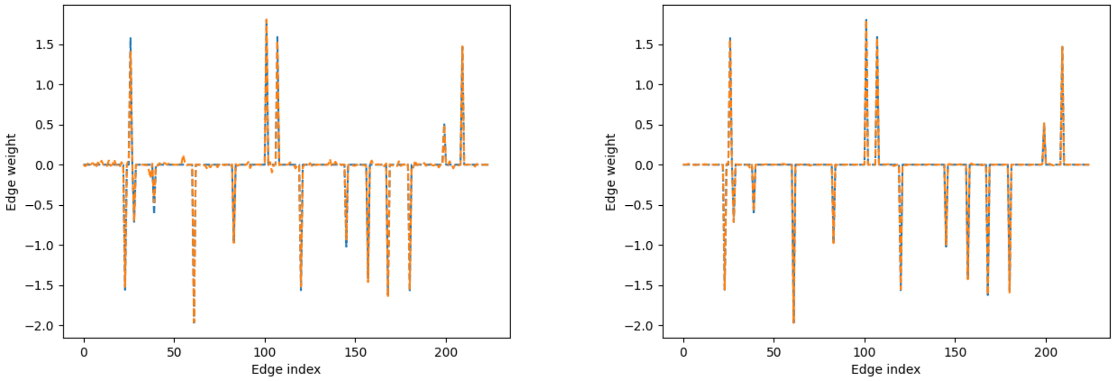


Figure 5.2: Graph estimation result with NSE $= 5.907 \times 10^{-3}$ at $T = 1000$ samples.

Figure 5.3: Graph estimation result with NSE $= 1.468 \times 10^{-3}$ as $T \to \infty$.

The estimation results at different sample sizes are depicted in Fig. 5.2 and Fig. 5.3. At $T = 1000$ samples, the NSE is 5.907×10^{-3} (Fig. 5.2), which improves to 1.468×10^{-3} as the number of samples approaches infinity (Fig. 5.3). In contrast to undirected graphs, our method does not completely converge to the true graph configuration. A notable aspect of these results is that our algorithm achieves commendable accuracy with relatively few samples, a benefit likely due to the integration of sparsity principles into our model.

5.3.2 Comparsion

Here, we compare our CovMatch method with the DAGs with no tears method [9] for scenarios involving infinite observations. The DAGs with no tears approach is specifically designed for DAGs and is capable of handling a variety of non-Gaussian distributions. However, it cannot address scenarios involving non-white Gaussian noise. Thus, we consider $\mathbf{e} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$. Here, we adopt the graph used in the DAG notears, which is generated when the random seed is set to zero.

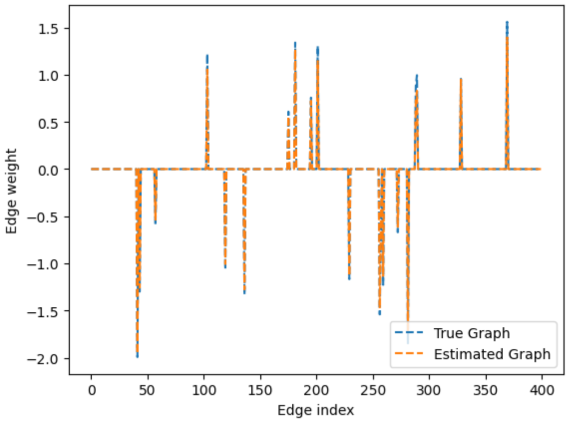


Figure 5.4: Results using DAGs with no tears method. NSE $= 6.727 \times 10^{-3}$.

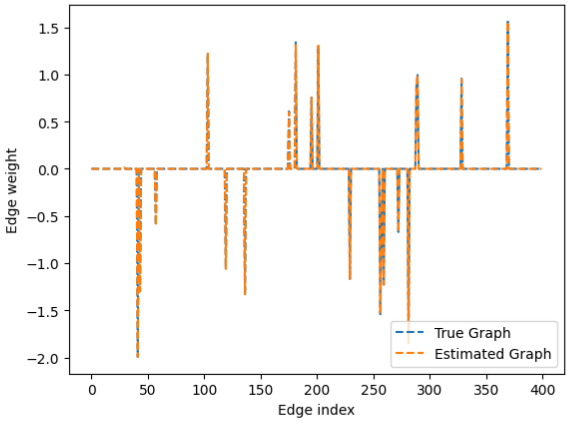


Figure 5.5: Results using our CovMatch method. NSE $= 3.944 \times 10^{-5}$.

As shown in Fig. 5.4 and Fig. 5.5, our CovMatch method significantly outperforms the DAGs with no tears method. Indeed, for DAGs, our results nearly match the true graph configuration. This suggests that graphs with cycles are inherently more challenging than acyclic graphs.

6 Extension to Polynomial Model

This chapter introduces an extension of our method to encompass polynomial relationships. We start by defining the problem and detailing its divergence from standard SEM formulations. We assume that the latent variables follow a zero mean white Gaussian distribution and that $\mathbf{S}$ is a hollow symmetric matrix. Within this framework, we derive a discrete quadratic programming (DQP) [34], for which we introduce a relaxation method Section 2.6 to find a solution.

We then broaden our discussion to scenarios accommodating a general Gaussian distribution, Finally, we will present a series of experiments designed to validate the effectiveness of our method.

6.1 Covariance Matching Optimization Problem

Assume $\mathbf{y} = \mathbf{H}\mathbf{x}$, where $\mathbf{H}$ is a polynomial of $\mathbf{S}$, denoted by $h(\mathbf{S})$ and $\mathbf{S}$ is a hollow symmetric matrix (for a SEM, $h(\mathbf{S}) = (\mathbf{I} - \mathbf{S})^{-1}$). Here, we use $\mathbf{x}$, which actually represents unmodeled inputs, analogous to the notation $\mathbf{e}$ commonly used in a SEM to denote unmodeled noise.

Using the following notation

$$\mathbf{\Sigma}_{\mathbf{y}} = \mathbb{E}\{\mathbf{y}\mathbf{y}^T\} = h(\mathbf{S})h(\mathbf{S})^\top = h(\mathbf{S})^2, \tag{6.1}$$

we will then try to match $\mathbf{C}_{\mathbf{y}}$ to $h^2(\hat{\mathbf{S}})$. This leads to the following problem

$$\begin{aligned} \hat{\mathbf{S}}^* = \arg\min_{\hat{\mathbf{S}}} \quad & \|h(\hat{\mathbf{S}})^2 - \mathbf{C}_{\mathbf{y}}\|_F^2 \\ \text{subject to} \quad & \operatorname{diag}(\hat{\mathbf{S}}) = 0, \\ & \hat{\mathbf{S}} = \hat{\mathbf{S}}^\top. \end{aligned} \tag{6.2}$$

To solve this, we replicate our EVD trick used in Subsection 3.2.2. Let $\hat{\mathbf{S}} = \hat{\mathbf{U}}\operatorname{diag}(\hat{\boldsymbol{\lambda}})\hat{\mathbf{U}}^\top$, which leads to $h^2(\hat{\mathbf{S}}) = \hat{\mathbf{U}}h^2(\operatorname{diag}(\hat{\boldsymbol{\lambda}}))\hat{\mathbf{U}}^\top$. Further, let the EVD of the sample covariance matrix be given by $\mathbf{C}_{\mathbf{y}} = \mathbf{U}_{\mathbf{y}}\operatorname{diag}(\boldsymbol{\lambda}_{\mathbf{y}})\mathbf{U}_{\mathbf{y}}^\top$. To match $\mathbf{C}_{\mathbf{y}}$ and $h^2(\hat{\mathbf{S}})$, we then set $\hat{\mathbf{U}} = \mathbf{U}_{\mathbf{y}}$ and $h^2(\operatorname{diag}(\hat{\boldsymbol{\lambda}})) = \operatorname{diag}(\boldsymbol{\lambda}_{\mathbf{y}})$.

Switching again the objective and constraint, we finally obtain the problem

$$\begin{aligned} \boldsymbol{\lambda}^* = \arg\min_{\hat{\boldsymbol{\lambda}}} \quad & \|(\mathbf{U}_{\mathbf{y}} \odot \mathbf{U}_{\mathbf{y}})\hat{\boldsymbol{\lambda}}\|_2^2 \\ \text{subject to} \quad & h^2(\operatorname{diag}(\hat{\boldsymbol{\lambda}})) - \operatorname{diag}(\boldsymbol{\lambda}_{\mathbf{y}}) = \mathbf{0}. \end{aligned} \tag{6.3}$$

The constraint basically represents a set of scalar polynomial constraints of the form $h^2(\hat{\lambda}_i) - \lambda_{i,\mathbf{y}} = 0$, $i = 1, 2, \ldots, N$, where $\hat{\lambda}_i$ ($\lambda_{i,\mathbf{y}}$) denotes the ith element of $\hat{\boldsymbol{\lambda}}$ ($\boldsymbol{\lambda}_\mathbf{y}$). Denoting the roots of the ith scalar polynomial as $\mathcal{C}_i = \{c_i^1, c_i^2, \ldots, c_i^{p_i}\}$ we can replace $h^2(\hat{\lambda}_i) - \lambda_{i,\mathbf{y}} = 0$ by $\hat{\lambda}_i \in \mathcal{C}_i$. Our proposed problem can finally be stated as

Optimization Problem 6.

$$\begin{aligned} \boldsymbol{\lambda}^* = \arg\min_{\hat{\boldsymbol{\lambda}}} \quad & \|(\mathbf{U_y} \odot \mathbf{U_y})\hat{\boldsymbol{\lambda}}\|_2^2 \\ \text{subject to } & \hat{\lambda}_i \in \mathcal{C}_i,\ i = 1, 2, \ldots, N. \end{aligned} \tag{6.4}$$

Remark 2. Let us explore how Optimization Problem 6 specializes to Optimization Problem 3. If we set $h(\mathbf{S}) = (\mathbf{I} - \mathbf{S})^{-1}$ in Optimization Problem 6, then all solutions to the equation $h^2(\operatorname{diag}(\hat{\boldsymbol{\lambda}})) - \operatorname{diag}(\boldsymbol{\lambda}_\mathbf{y}) = \mathbf{0}$ can be expressed as $\hat{\boldsymbol{\lambda}} = \operatorname{diag}(\boldsymbol{\lambda}_\mathbf{y}^{-1/2})\hat{\mathbf{q}} - \mathbf{1}$, where $\hat{\mathbf{q}} \in \{-1, 1\}^{N \times 1}$. Therefore, the objective of Optimization Problem 6 can be rewritten as $(\mathbf{U_y} \odot \mathbf{U_y})(\operatorname{diag}(\boldsymbol{\lambda}_\mathbf{y}^{-1/2})\hat{\mathbf{q}} - \mathbf{1})$, which, due to the property $(\mathbf{U_y} \odot \mathbf{U_y})\mathbf{1} = \mathbf{1}$, becomes identical to the problem defined in Optimization Problem 3. Thus, Optimization Problem 6 can be viewed as an extension of Optimization Problem 3.

Since

$$\|(\mathbf{U_y} \odot \mathbf{U_y})\hat{\boldsymbol{\lambda}}\|_2^2 = \hat{\boldsymbol{\lambda}}^\top (\mathbf{U_y} \odot \mathbf{U_y})^\top (\mathbf{U_y} \odot \mathbf{U_y})\hat{\boldsymbol{\lambda}},$$

this problem can be solved using the method discussed in Section 2.6.

6.2 Extension of the Latent Variable Distribution

The previous assumption that $\mathbf{x} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ is overly idealistic and significantly limits the applicability of the method. To address these limitations and apply the method to more realistic scenarios, we propose a method for a general Gaussian distribution.

Consider the latent variables $\mathbf{x}$ to be normally distributed as $\mathbf{x} \sim \mathcal{N}(\boldsymbol{\mu}_\mathbf{x}, \boldsymbol{\Sigma}_\mathbf{x})$ ($\boldsymbol{\mu}_\mathbf{x} \neq \mathbf{0}$), where both $\boldsymbol{\mu}_\mathbf{x}$ and $\boldsymbol{\Sigma}_\mathbf{x}$ are known. For $\mathbf{y}$, the sample mean and covariance are denoted by $\boldsymbol{m}_\mathbf{y}$ and $\mathbf{C_y}$, respectively.

The first step involves estimating $\mathbf{H}$ in the relation $\mathbf{y} = \mathbf{Hx}$, where $\mathbf{H} = h(\mathbf{S})$ is a polynomial of $\mathbf{S}$. In a second step we then estimate $\mathbf{S}$ from $\mathbf{H}$.

To estimate $\mathbf{H}$, we consider the relation $\boldsymbol{\Sigma}_\mathbf{y} = \mathbf{H}\boldsymbol{\Sigma}_\mathbf{x}\mathbf{H}^\top$. Hence, we could match $\hat{\mathbf{H}}\boldsymbol{\Sigma}_\mathbf{x}\hat{\mathbf{H}}^\top$ with $\mathbf{C_y}$.

Solving this matching problem is challenging though. As an alternative, observe that $(\mathbf{H}\boldsymbol{\Sigma}_\mathbf{x})^2 = \mathbf{H}\boldsymbol{\Sigma}_\mathbf{x}\mathbf{H}\boldsymbol{\Sigma}_\mathbf{x} = \mathbf{C_y}\boldsymbol{\Sigma}_\mathbf{x}$. This allows us to match $(\hat{\mathbf{H}}\boldsymbol{\Sigma}_\mathbf{x})^2$ with $\mathbf{C_y}\boldsymbol{\Sigma}_\mathbf{x}$ which is similar to (3.12) and (6.2), where we matched $\hat{\mathbf{H}}^2$ with $\mathbf{C_y}$.

Assume the EVD of $\hat{\mathbf{H}}\boldsymbol{\Sigma}_\mathbf{x}$ is given by[1] $\hat{\mathbf{H}}\boldsymbol{\Sigma}_\mathbf{x} = \hat{\mathbf{U}} \operatorname{diag}(\hat{\boldsymbol{\lambda}})\hat{\mathbf{U}}^{-1}$, then $(\hat{\mathbf{H}}\boldsymbol{\Sigma}_\mathbf{x})^2 = \hat{\mathbf{U}} \operatorname{diag}(\hat{\boldsymbol{\lambda}})^2\hat{\mathbf{U}}^{-1}$. For the EVD of $\mathbf{C_y}\boldsymbol{\Sigma}_\mathbf{x}$, let $\mathbf{C_y}\boldsymbol{\Sigma}_\mathbf{x} = \mathbf{U_{yx}} \operatorname{diag}(\boldsymbol{\lambda}_\mathbf{yx})\mathbf{U}_\mathbf{yx}^{-1}$. Setting

[1] Here, the notation $\mathbf{U}$ is again primarily used to align with previous notation and does not imply that $\mathbf{U}$ is unitary.

$\hat{\mathbf{U}} = \mathbf{U_{yx}}$ and $\hat{\boldsymbol{\lambda}}^2 = \boldsymbol{\lambda}_{\mathbf{yx}}$ concludes the covariance matching but introduces a sign ambiguity. To resolve this, introduce $\hat{\mathbf{q}} \in \{-1, 1\}^{N\times 1}$, and define $\hat{\boldsymbol{\lambda}} = \operatorname{diag}(\boldsymbol{\lambda}_{\mathbf{yx}})\hat{\mathbf{q}}$.

We can then write $\hat{\mathbf{H}}$ as

$$\hat{\mathbf{H}} = \mathbf{U_{yx}}\mathbf{D}_{\mathbf{yx}}^{1/2}\operatorname{diag}(\hat{\mathbf{q}})\mathbf{U}_{\mathbf{yx}}^{-1}\mathbf{C}_{\mathbf{x}}^{-1}. \tag{6.5}$$

Now we want to match the mean, that is match $\hat{\mathbf{H}}\boldsymbol{\mu}_{\mathbf{x}}$ to $\mathbf{m_y}$. Notice that

$$\hat{\mathbf{H}}\boldsymbol{\mu}_{\mathbf{x}} = \mathbf{U_{yx}}\mathbf{D}_{\mathbf{yx}}^{1/2}\operatorname{diag}(\hat{\mathbf{q}})\mathbf{U}_{\mathbf{yx}}^{-1}\mathbf{C}_{\mathbf{x}}^{-1}\boldsymbol{\mu}_{\mathbf{x}} = \mathbf{U_{yx}}\mathbf{D}_{\mathbf{yx}}^{1/2}\operatorname{diag}(\mathbf{U}_{\mathbf{yx}}^{-1}\mathbf{C}_{\mathbf{x}}^{-1}\boldsymbol{\mu}_{\mathbf{x}})\hat{\mathbf{q}} \tag{6.6}$$

Let $\mathbf{W} = \mathbf{U_{yx}}\mathbf{D}_{\mathbf{yx}}^{1/2}\operatorname{diag}(\mathbf{U}_{\mathbf{yx}}^{-1}\mathbf{C}_{\mathbf{x}}^{-1}\boldsymbol{\mu}_{\mathbf{x}})$, the resulting optimization problem is given by **Optimization Problem 7.**

$$\min_{\hat{\mathbf{q}} \in \{1,1\}^{N\times 1}} \|\mathbf{W}\hat{\mathbf{q}} - \mathbf{m_y}\|_F^2 \tag{6.7}$$

This is again a BQP and can be solved by the methd in Section 2.5

After estimating $\mathbf{H}$, the next step involves estimating $\mathbf{S}$ from $\mathbf{H}$, which is actually matching $h(\hat{\mathbf{S}})$ to $\hat{\mathbf{H}}$. This procedure is analogous to our previous approach of matching $h^2(\hat{\mathbf{S}})$ to $\mathbf{C_y}$ and will not be elaborated further in this thesis.

Remark 3. One might wonder why we do not utilize the symmetric properties in our optimization problem, but instead use the mean as the basis of our optimization. The following lemma, with a proof provided by Dan Fulea [35], clarifies this choice.

Lemma 4. Let $\mathbf{H}$ and $\mathbf{C_x}$ be symmetric matrices, with $\mathbf{C_x}$ being invertible, and suppose $\mathbf{HC_x}$ has an EVD given by $\mathbf{UDU}^{-1}$. If $\mathbf{D}$ has all distinct diagonal values, then, for any diagonal matrix $\mathbf{D}'$, $\mathbf{UD}'\mathbf{U}^{-1}$ is always symmetric.

Given the lemma discussed, we recognize that the condition of symmetry for the matrix $\mathbf{S}$ is almost naturally fulfilled. In other words, this implies that regardless of the choice of $\mathbf{q}$ in Optimization Problem 7, for a large portion of $\mathbf{S}$, the estimated $\mathbf{H}^*$ is invariably symmetric. Thus, the symmetry of $\mathbf{H}$ is almost redundant information in this context.

Remark 4. Here, we assume that $\hat{\mathbf{H}}\mathbf{C_x}$ admits a real-valued EVD. Actually, a real-valued EVD exists if $\hat{\mathbf{H}}\mathbf{C_x}$ is a full rank matrix with distinct eigenvalues. Here, EVD exists because it has distinct eigenvalues [36]. It's real because $\lambda(\hat{\mathbf{H}}\mathbf{C_x}) = \lambda(\mathbf{C}_{\mathbf{x}}^{\frac{1}{2}}\hat{\mathbf{H}}\mathbf{C}_{\mathbf{x}}^{\frac{1}{2}})$ and all eigenvalues of $\mathbf{C}_{\mathbf{x}}^{\frac{1}{2}}\hat{\mathbf{H}}\mathbf{C}_{\mathbf{x}}^{\frac{1}{2}}$ are real. This is a corollary from [37]

6.3 Experiments

6.3.1 Simulation Experiment

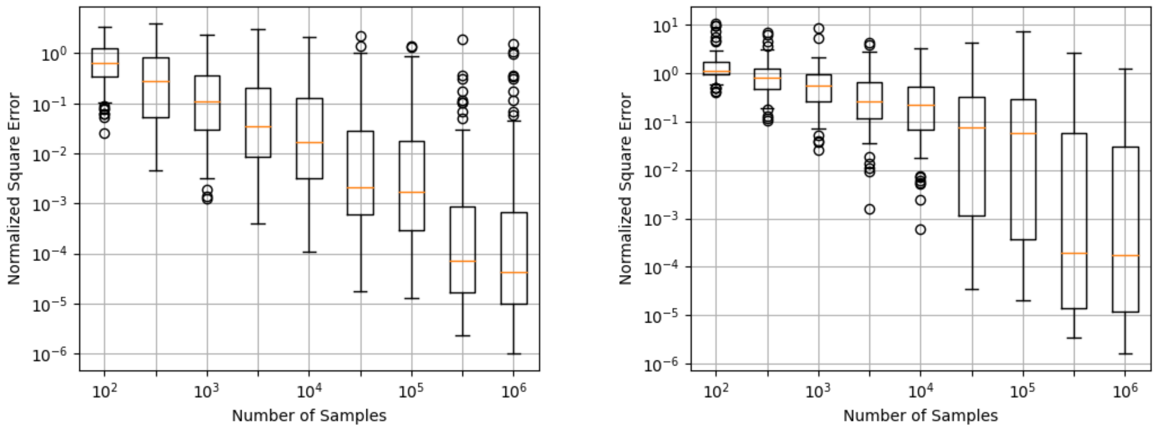
Here we consider 100 graphs with 20 nodes and 40 edges with edge weights belonging to $[-2, -0.1] \cup [0.1, 2]$.

The function $h(\lambda)$, representing the spectral filter applied to the Laplacian eigenvalues, was chosen to be a cubic polynomial given by:

$$h(\lambda) = h_3\lambda^3 + h_2\lambda^2 + h_1\lambda + h_0,$$

where h_0, h_1, h_2, and h_3 are coefficients uniformly randomly drawn from the interval $[-1, 1]$.

To configure the covariance $\mathbf{\Sigma}_{\mathbf{x}}$, we first generated a random sqaure root matrix, denoted as $\mathbf{\Sigma}_{\text{sqrt}}$, which is an $N \times N$ matrix with each element uniformly distributed in the range [-1, 1]. The matrix $\mathbf{\Sigma}_{\mathbf{x}}$ was then formed by $\mathbf{\Sigma}_{\text{sqrt}}\mathbf{\Sigma}_{\text{sqrt}}^{\top}$, ensuring that $\mathbf{\Sigma}_{\mathbf{x}}$ is symmetric and positive definite, which is crucial for the stability of the graph-based processes studied. For the latent mean $\boldsymbol{\mu}_{\mathbf{x}}$, each entry is uniformly at random drawn from the interval $[-1, 1]$.



(a) Boxplot of average NSE for $\hat{\mathbf{H}}$ for different sample sizes

(b) Boxplot of average NSE for $\hat{\mathbf{S}}$ for different sample sizes

Figure 6.1: As $T \to \infty$, the average NSE decreases to below the order of 10^{-5}.

As shown in the Fig. 6.1b, with the increase in the number of samples, the NSE for most graphs reduces to very low levels, although it remains relatively high for a few. When we let $T \to \infty$, the NSE drops below 10^{-5} for all graphs, demonstrating that our method can nearly perfectly estimate the graph structure given a sufficient number of samples.

6.3.2 Real Data Experiment

We also compare our approach with network deconvolution (referred to as NetDeconv) [15], which similarly involves estimating $\mathbf{S}$ from $\mathbf{H}$. In this experiment, each node within the network corresponds to an amino acid residue, and the edges denote mutual information, reflecting co-variation among residues across multiple sequence alignments that include 2,000 to 72,000 sequences. Our objective is to deduce structural constraints among amino acid pairs to aid in predicting protein structures.

We employ a relatively straightforward polynomial $h(x) = \frac{1}{80}(x^3 + 2x^2 + 4x)$, whereas NetDeconv approximates $h(x) = \frac{x}{1-x}$. Terms such as "1wvn" shown in Fig. 6.2 represent different protein labels. The figure illustrates that in various cases, our results outperform those of NetDeconv.

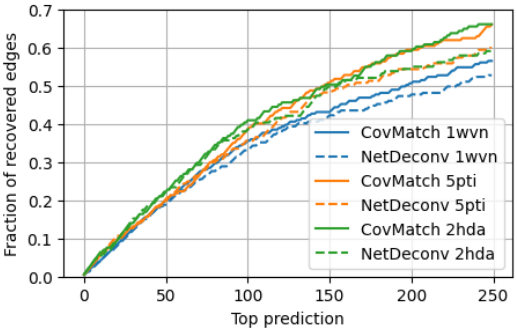


Figure 6.2: Real contact edge recovery as a function of the number of edges considered.

7 Conclusion and Future work

7.1 Conclusions

In this thesis, as shown in Table 7.1, we have systematically addressed a broad class of problems in graph topology identification (GTI) within the framework of covariance matching. Our method introduces a straightforward notion: the desire for our estimates to reproduce the observed covariance. Our final experimental results demonstrate that our approach is more powerful than many existing methods and shows substantial potential for addressing more complex issues in future extensions. Overall, the covariance matching approach introduces new possibilities and directions within the field of GTI.

Table 7.1: Overview of Our Method's Applicability

Model Type	**Is Directed**	**Is Sparse**	**Latent Variable Distribution**	**Applicability**
SEM	No	–	$\mathcal{N}(0, \mathbf{\Sigma})$	Applicable
SEM	Yes	No	$\mathcal{N}(0, \mathbf{\Sigma})$	Unsolvable
SEM	Yes	Yes	$\mathcal{N}(0, \mathbf{\Sigma})$	Applicable
PM	No	–	$\mathcal{N}(\boldsymbol{\mu}, \mathbf{\Sigma}), \boldsymbol{\mu} \neq \mathbf{0}$	Applicable
PM	No	–	$\mathcal{N}(0, \mathbf{I})$	Applicable
PM	Yes	–	–	Unknown

Specifically, we started with undirected graphs, proving that for a large portion of graphs, our method can converge to the true graph structure, a feat unachievable by other methods based solely on signal matching. We studied this method for white as well as non-white Gaussian latent variables.

Furthermore, we ventured into the theoretically challenging territory of directed graphs, adhering still to the covariance matching approach. First, we consider a convex relaxation based approach for the case of white Gaussian latent variables and a sparse graph. We then proposed a unified method capable of broadly addressing non-white Gaussian latent variables and sparse graphs, which yielded surprisingly positive results. The feasibility of our approach for cyclic graphs, seldom discussed in the literature, proved more effective than some existing methods limited only to directed acyclic graphs (DAGs).

Finally, we extended our methodology to polynomial models, broadening the potential applications of our approach to more complex scenarios.

In conclusion, our work not only advances the theoretical and practical aspects of GTI but also sets the stage for future explorations that could further exploit the intricacies of covariance matching in increasingly sophisticated models.

7.2 Future Work

In completing this thesis, I became aware of numerous potential directions for further research, some of which are already taking shape. I wish to candidly write these ideas here, hoping they may guide future research along these lines. For each chapter, I have highlighted a problem that still needs to be solved.

Firstly, while our solution for a SEM based on undirected graphs is mathematically elegant and conceptually straightforward, it still harbors a fundamental issue: our complete reliance on the fact that the spectrum (or, more specifically, the eigenvectors) for finite samples is the same as the theoretical spectrum. This trust is not entirely feasible, leading to performance issues for fewer observations, as discussed in Subsection 3.4.4. Early on, I recognized this problem, which seemed formidable but not insurmountable. A potential solution could leverage the beautiful mathematical property that the space of unitary matrices is closed under matrix multiplication. If we have an inaccurate spectrum $\mathbf{U_y}$, we could estimate it as $\hat{\mathbf{U}}_\mathbf{y} = \mathbf{U_y}\hat{\mathbf{U}}$, where $\hat{\mathbf{U}}$ is a unitary matrix ideally equal to the identity matrix. This problem is akin to the one described in the unified approach of Chapter 6.

Secondly, I discovered that the unitary optimization problem, employed in the unified approach, could be addressed via gradient descent. Indeed, computing the gradient for a unitary matrix equates to solving an optimization problem concerning a constrained skew-symmetric matrix, which is nearly a linear problem. This discovery could replace our current iterative method.

Thirdly, the reason behind the success of the unified approach remains unclear. The method has proved more successful than anticipated, leading me to hypothesize that a significant class of graphs exists where our method is effective, or the error is bounded. I believe this issue depends on the resolution of the second point.

Fourthly, we could think about extending our methods to dynamic scenarios. There is a trivial approach to this: one can always obtain a covariance matrix within a window and allow only limited changes from one window to the next.

Fifthly, regarding the polynomial model, I am convinced that a more elegant solution exists. My method initially solves an equation $h^2(\mathrm{diag}(\hat{\boldsymbol{\lambda}})) - \mathrm{diag}(\boldsymbol{\lambda}_\mathbf{y}) = \mathbf{0}$; however, due to errors, the equation can become rootless, rendering the problem unsolvable. We considered alternatives to tackle this issue like gradient descent, but found them unsatisfactory due to high order derivatives leading to significant numerical instability.

Bibliography

[1] Xiaowen Dong, Dorina Thanou, Michael Rabbat, and Pascal Frossard. Learning graphs from data: A signal representation perspective. *IEEE Signal Processing Magazine*, 36(3):44–63, 2019.

[2] Andrei Buciulea, Jiaxi Ying, Antonio G Marques, and Daniel P Palomar. Polynomial graphical lasso: Learning edges from gaussian graph-stationary signals. *arXiv preprint arXiv:2404.02621*, 2024.

[3] Jiaying Liu, Feng Xia, Lei Wang, Bo Xu, Xiangjie Kong, Hanghang Tong, and Irwin King. Shifu2: A network representation learning based model for advisor-advisee relationship mining. *IEEE Transactions on Knowledge and Data Engineering*, 33(4):1763–1777, 2019.

[4] AR Mclntosh and Francisco Gonzalez-Lima. Structural equation modeling and its application to network analysis in functional brain imaging. *Human brain mapping*, 2(1-2):2–22, 1994.

[5] Judea Pearl. Graphs, causality, and structural equation models. *Sociological Methods & Research*, 27(2):226–284, 1998.

[6] Alberto Natali, Elvin Isufi, Mario Coutino, and Geert Leus. Online graph learning from time-varying structural equation models. In *2021 55th Asilomar Conference on Signals, Systems, and Computers*, pages 1579–1585, 2021.

[7] Juan Andrés Bazerque, Brian Baingana, and Georgios B Giannakis. Identifiability of sparse structural equation models for directed and cyclic networks. In *2013 IEEE Global Conference on Signal and Information Processing*, pages 839–842. IEEE, 2013.

[8] Jonas Peters and Peter Bühlmann. Identifiability of gaussian structural equation models with equal error variances. *Biometrika*, 101(1):219–228, 2014.

[9] Xun Zheng, Bryon Aragam, Pradeep K Ravikumar, and Eric P Xing. Dags with no tears: Continuous optimization for structure learning. *Advances in neural information processing systems*, 31, 2018.

[10] Seyed Saman Saboksayr, Gonzalo Mateos, and Mariano Tepper. Block successive convex approximation for concomitant linear dag estimation. In *2024 IEEE 13rd Sensor Array and Multichannel Signal Processing Workshop (SAM)*, pages 1–5. IEEE, 2024.

[11] Gonzalo Mateos, Santiago Segarra, Antonio G Marques, and Alejandro Ribeiro. Connecting the dots: Identifying network structure via graph signal processing. *IEEE Signal Processing Magazine*, 36(3):16–43, 2019.

[12] Alberto Natali, Elvin Isufi, Mario Coutino, and Geert Leus. Learning time-varying graphs from online data. *IEEE Open Journal of Signal Processing*, 3:212–228, 2022.

[13] Rasoul Shafipour, Santiago Segarra, Antonio G Marques, and Gonzalo Mateos. Identifying the topology of undirected networks from diffused non-stationary graph signals. *IEEE Open Journal of Signal Processing*, 2:171–189, 2021.

[14] Santiago Segarra, Antonio G Marques, Gonzalo Mateos, and Alejandro Ribeiro. Network topology inference from spectral templates. *IEEE Transactions on Signal and Information Processing over Networks*, 3(3):467–483, 2017.

[15] Soheil Feizi, Daniel Marbach, Muriel Médard, and Manolis Kellis. Network deconvolution as a general method to distinguish direct dependencies in networks. *Nature biotechnology*, 31(8):726–733, 2013.

[16] Koki Yamada, Yuichi Tanaka, and Antonio Ortega. Time-varying graph learning based on sparseness of temporal variation. In *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5411–5415. IEEE, 2019.

[17] David Alejandro Jimenez-Sierra, David Alfredo Quintero-Olaya, Juan Carlos Alvear-Munoz, Hernan Dario Benitez-Restrepo, Juan Felipe Florez-Ospina, and Jocelyn Chanussot. Graph learning based on signal smoothness representation for homogeneous and heterogeneous change detection. *IEEE Transactions on Geoscience and Remote Sensing*, 60:1–16, 2022.

[18] Beresford N Parlett. *The symmetric eigenvalue problem.* SIAM, 1998.

[19] Kenneth A Bollen. *Structural equations with latent variables.* John Wiley & Sons, 2014.

[20] Rex B Kline. *Principles and practice of structural equation modeling.* Guilford publications, 2023.

[21] Xiaodong Cai, Juan Andrés Bazerque, and Georgios B Giannakis. Inference of gene regulatory networks with sparse structural equation models exploiting genetic perturbations. *PLoS computational biology*, 9(5):e1003068, 2013.

[22] Georgios B Giannakis, Yanning Shen, and Georgios Vasileios Karanikolas. Topology identification and learning over graphs: Accounting for nonlinearities and dynamics. *Proceedings of the IEEE*, 106(5):787–807, 2018.

[23] Zhi-Quan Luo, Wing-Kin Ma, Anthony Man-Cho So, Yinyu Ye, and Shuzhong Zhang. Semidefinite relaxation of quadratic optimization problems. *IEEE Signal Processing Magazine*, 27(3):20–34, 2010.

[24] Michel X Goemans and David P Williamson. Improved approximation algorithms for maximum cut and satisfiability problems using semidefinite programming. *Journal of the ACM (JACM)*, 42(6):1115–1145, 1995.

[25] Zhi-Quan Luo, Wing-Kin Ma, Anthony Man-Cho So, Yinyu Ye, and Shuzhong Zhang. Semidefinite relaxation of quadratic optimization problems. *IEEE Signal Processing Magazine*, 27(3):20–34, 2010.

[26] Gurobi Optimization, LLC. Gurobi Optimizer Reference Manual, 2024.

[27] Christoph Buchheim and Angelika Wiegele. Semidefinite relaxations for non-convex quadratic mixed-integer programming. *Mathematical Programming*, 141:435–452, 2013.

[28] Keith Conrad. The minimal polynomial and some applications. *matrix*, 1:1, 2014.

[29] Kaare Brandt Petersen, Michael Syskind Pedersen, et al. The matrix cookbook. *Technical University of Denmark*, 7(15):510, 2008.

[30] Michael Grant and Stephen Boyd. CVX: Matlab software for disciplined convex programming, version 2.1. https://cvxr.com/cvx, March 2014.

[31] Jean Gallier et al. The schur complement and symmetric positive semidefinite (and definite) matrices. *Penn Engineering*, pages 1–12, 2010.

[32] DCP ruleset. https://web.cvxr.com/cvx/doc/dcp.html. Accessed: date-of-access.

[33] Peter H Schönemann. A generalized solution of the orthogonal procrustes problem. *Psychometrika*, 31(1):1–10, 1966.

[34] Tiantian Nie. *Quadratic Programming with Discrete Variables.* North Carolina State University, 2016.

[35] Fulea Dan. Prove a matrix is always symmetric. Mathematics Stack Exchange. URL: https://math.stackexchange.com/q/4952162 (version: 2024-07-29).

[36] Mathematicing (https://math.stackexchange.com/users/227967/mathematicing). Distinct eigenvalues implies $a \in \mathbb{R}^{n\times n}$ is diagonalisable. Mathematics Stack Exchange. URL:https://math.stackexchange.com/q/3042187 (version: 2018-12-16).

[37] icurays1 (https://math.stackexchange.com/users/49070/icurays1). The eigenvalues of the product of a positive definite and a symmetric matrix. Mathematics Stack Exchange. URL:https://math.stackexchange.com/q/522961 (version: 2013-10-11).